

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC VÀ KỸ THUẬT THÔNG TIN

ĐẶNG HOÀNG QUÂN
NGUYỄN ĐỨC DUY ANH

KHÓA LUẬN TỐT NGHIỆP
TẬN DỤNG DEEP LEARNING ĐỂ SỐ HÓA CHỮ VIẾT
TAY VIỆT NAM CŨ CHO LUU TRỮ TÀI LIỆU LỊCH SỬ
LEVERAGE DEEP LEARNING TO DIGITIZE OLD
VIETNAMESE HANDWRITTEN FOR HISTORICAL
DOCUMENT ARCHIVING

CỦ NHÂN NGÀNH KHOA HỌC DỮ LIỆU

TP. HỒ CHÍ MINH, 2022

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC VÀ KỸ THUẬT THÔNG TIN

ĐẶNG HOÀNG QUÂN – 18520339

NGUYỄN ĐỨC DUY ANH – 18520455

KHÓA LUẬN TỐT NGHIỆP

**TẬN DỤNG DEEP LEARNING ĐỂ SỐ HÓA CHỮ VIẾT
TAY VIỆT NAM CŨ CHO LUU TRỮ TÀI LIỆU LỊCH SỬ**
**LEVERAGE DEEP LEARNING TO DIGITIZE OLD
VIETNAMESE HANDWRITTEN FOR HISTORICAL
DOCUMENT ARCHIVING**

CỦ NHÂN NGÀNH KHOA HỌC DỮ LIỆU

GIẢNG VIÊN HƯỚNG DẪN

TS. ĐỖ TRỌNG HỢP

THS. TẠ THU THỦY

TP. HỒ CHÍ MINH, 2022

THÔNG TIN HỘI ĐỒNG CHẤM KHÓA LUẬN TỐT NGHIỆP

Hội đồng chấm khóa luận tốt nghiệp, thành lập theo Quyết định số 124/QĐ-DHCNTT ngày 04 tháng 3 năm 2022 của Hiệu trưởng Trường Đại học Công nghệ Thông tin.

LỜI CẢM ƠN

Vô số sự đồng hành và ủng hộ cho những nỗ lực của chúng tôi trong bài luận này. Lời đầu tiên, xin được dành lời cảm ơn chân thành nhất đến các thầy cô hướng dẫn. Chúng em trân trọng những lời khuyên hữu ích và phản hồi vô giá, đôi khi là cả những trả lời tin nhắn vào đêm muộn và sáng sớm của TS. Đỗ Trọng Hợp đã giúp nhóm có được những cái nhìn lần hướng đi đúng đắn. Chúng em cũng muốn cảm ơn ThS. Tạ Thu Thủy vì sự giúp đỡ và hỗ trợ của cô trong suốt chặng đường vừa qua. Chúng em tự hào và biết ơn những khoảng thời gian được làm việc cùng đó.

Nhóm cũng xin gửi lời cảm ơn đến các thầy cô Khoa Khoa học và Kỹ thuật Thông tin nói riêng cũng như các thầy cô Trường Đại học Công nghệ Thông tin nói chung đã tận tâm chỉ dạy, truyền đạt những kiến thức quý báu cũng như các kỹ năng cần thiết để nhóm có thể đạt được những thành công nhất định trong tương lai.

Cảm ơn các bạn trong nhóm gán nhãn, những người đã hào phóng hy sinh một phần thời gian biểu để tham gia vào nghiên cứu của chúng tôi cũng như giúp nhóm có thể hoàn thành bộ dữ liệu thật tốt, từ đó góp phần biến đề tài này thành hiện thực.

Bên cạnh đó, nhóm cũng muốn gửi lời cảm ơn đến bạn Nguyễn Ngọc Thịnh, ngành Đông Phương học, Trường Đại học Khoa học Xã hội và Nhân văn đã giúp chúng tôi giải đáp những thắc mắc liên quan đến chữ Hán-Nôm trong đề tài này.

Cảm ơn thầy Nguyễn Đạt Phi, người sáng lập kênh Hùng Ca Sử Việt đã truyền cho chúng tôi ngọn lửa tình yêu với lịch sử dân tộc. Những câu chuyện về cha ông được kể lại qua giọng đọc truyền cảm của thầy đã trở thành một món ăn tinh thần không thể thiếu, từ đó mà chúng tôi biết ơn người xưa và quý trọng sự ám no bây giờ. Đây đồng thời cũng là động lực đưa nhóm đến với đề tài này.

Và lời cảm ơn cuối cùng xin chân thành gửi đến gia đình và bạn bè vì đã luôn bên cạnh động viên và ủng hộ, tạo điều kiện tinh thần và vật chất cho chúng tôi trong quá trình học tập và nghiên cứu.

Nhóm tác giả

MỤC LỤC

LỜI CẢM ƠN	4
MỤC LỤC	5
DANH MỤC HÌNH	9
DANH MỤC BẢNG	12
DANH MỤC TỪ VIẾT TẮT	13
TÓM TẮT KHÓA LUẬN	1
Chương 1. MỞ ĐẦU	2
1.1. Đặt vấn đề	2
1.2. Lý do chọn đề tài	3
1.3. Mục tiêu khóa luận	4
1.4. Đối tượng và phạm vi nghiên cứu	4
1.5. Các nội dung chính	5
Chương 2. TỔNG QUAN	6
2.1. Giới thiệu đề tài	6
2.2. Tính ứng dụng của đề tài	7
2.3. Thách thức	8
Chương 3. NGHIÊN CỨU LIÊN QUAN	9
3.1. Tình hình nghiên cứu trên thế giới	9
3.2. Tình hình nghiên cứu trong nước	11
Chương 4. CƠ SỞ LÝ THUYẾT	13
4.1. Nhận dạng ký tự quang học (OCR)	13
4.1.1. Các khái niệm cơ bản	14
4.1.2. Phân loại hình ảnh chứa văn bản	15
4.1.3. OCR và Học sâu	17
4.1.4. Các bước triển khai chính	18
4.1.5. Một số dataset cho văn bản phi cấu trúc	19
4.1.6. Một số công cụ mã nguồn mở	20
4.2. Các thành phần tính toán chính	21

4.2.1.	Mạng Nơ-ron Tích chập (CNN)	21
4.2.1.1.	Các khái niệm cơ bản.....	21
4.2.1.2.	Chuẩn hóa Batch (Batch Normalization).....	23
4.2.1.3.	Kết nối tắt (Skip connection)	24
4.2.2.	Phân vùng ảnh (Image Segmentation).....	25
4.2.2.1.	Các khái niệm cơ bản.....	25
4.2.2.2.	Ý tưởng từ mạng FCN	26
4.2.2.3.	Mạng U-Net	26
4.2.3.	Mạng Nơ-ron Hồi tiếp (RNN)	27
4.2.3.1.	Các khái niệm cơ bản.....	27
4.2.3.2.	Embedding từ (Word embedding)	28
4.2.3.3.	Nút Hồi tiếp có Cổng (GRU)	30
4.2.3.4.	Mạng Nơ-ron Hồi tiếp 2 chiều.....	31
4.2.3.5.	Mô hình chuỗi sang chuỗi (Seq2Seq).....	32
4.2.4.	Cơ chế Tập trung (Attention Mechanism)	33
4.2.4.1.	Khởi nguồn	33
4.2.4.2.	Các tính toán chính	34
4.2.4.3.	Seq2Seq sử dụng Cơ chế Tập trung.....	35
4.2.4.4.	Tự tập trung (Self-Attention).....	35
4.2.4.5.	Kiến trúc Transformer	36
Chương 5.	BỘ DỮ LIỆU NomNaOCR	39
5.1.	Khái quát chung	39
5.2.	Thu thập dữ liệu.....	40
5.3.	Quy trình gán nhãn	42
5.3.1.	Xây dựng hướng dẫn (Guideline).....	43
5.3.2.	Gán nhãn tự động (Auto annotation).....	44
5.3.3.	Quy trình đánh giá	45
5.3.4.	Triển khai thực tế.....	46
5.4.	Các khó khăn cùng hướng xử lý	48

5.5.	Phân tích và chia dữ liệu.....	51
5.6.	Bộ dữ liệu Synthetic Nom String	54
Chương 6.	CÁC PHƯƠNG PHÁP TIẾP CẬN	56
6.1.	Khởi nguồn và lý do tiếp cận bằng Học sâu.....	56
6.2.	Phát hiện văn bản (Text Detection)	57
6.2.1.	Tiếp cận theo Regression-based với EAST.....	58
6.2.2.	Tiếp cận theo Segmentation-based với DBNet	60
6.3.	Nhận dạng văn bản (Text Recognition).....	63
6.3.1.	Tiếp cận theo hướng sinh mô tả cho ảnh.....	63
6.3.1.1.	Kiến trúc Injection và Merging.....	64
6.3.1.2.	Kiến trúc dựa trên Cơ chế Tập trung	65
6.3.2.	Mạng Nơ-ron Hồi tiếp Tích chập (CRNN)	67
6.3.2.1.	Kiến trúc	67
6.3.2.2.	CTC Loss	69
6.3.3.	Tiếp cận theo hướng Seq2Seq trong dịch máy.....	70
6.3.4.	Các mô hình TransformerOCR	72
Chương 7.	CÀI ĐẶT THỬ NGHIỆM	74
7.1.	Triển khai cho bài toán Text Detection	74
7.2.	Triển khai cho bài toán Text Recognition	75
7.2.1.	Các giai đoạn huấn luyện	75
7.2.2.	CNN backbone	75
7.2.3.	Cài đặt phần Xử lý ngôn ngữ	77
7.2.4.	Thuật toán tối ưu (Optimizer)	77
7.2.5.	Các thông số khác.....	78
7.2.6.	Thử nghiệm với các Kết nối tắt	79
Chương 8.	ĐÁNH GIÁ VÀ KẾT QUẢ.....	80
8.1.	Phương pháp đánh giá	80
8.1.1.	Metrics đánh giá Text Detection và End-to-End.....	80
8.1.2.	Metrics đánh giá với riêng Text Recognition.....	82

8.2. Kết quả thử nghiệm	83
8.2.1. Kết quả bài toán Text Detection	83
8.2.1.1. Kết quả tổng quan	84
8.2.1.2. Kết quả theo từng tác phẩm	84
8.2.2. Kết quả bài toán Text Recognition	86
8.2.2.1. Kết quả giai đoạn Pre-training	86
8.2.2.2. Kết quả Fine-tuning và Retraining	87
8.2.2.3. Kết quả các ngưỡng 10 ký tự	89
8.2.3. Kết quả End-to-End	92
8.2.3.1. Kết quả trên toàn bộ ảnh	92
8.2.3.2. Kết quả chi tiết trên thơ và văn xuôi	93
8.2.3.3. Nhận định	95
8.3. Phân tích lỗi	95
8.3.1. Phân tích lỗi cho bài toán Text Detection	95
8.3.2. Phân tích lỗi cho bài toán Text Recognition	97
Chương 9. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	99
9.1. Tổng kết	99
9.2. Hướng nghiên cứu trong tương lai	100
TÀI LIỆU THAM KHẢO	101

DANH MỤC HÌNH

Hình 4.1. OCR cho văn bản Hán-Nôm	13
Hình 4.2. Ảnh chứa văn bản có cấu trúc (Nguồn [15]).....	15
Hình 4.3. Ảnh chứa Graphic text và Scene text (Nguồn [16])......	16
Hình 4.4. Ảnh chứa Synthetic text (Nguồn [17]).....	16
Hình 4.5. Ảnh chứa văn bản lai giữa có và phi cấu trúc trong NomNaOCR.....	17
Hình 4.6. Pipeline với 2 bài toán con chính trong OCR	19
Hình 4.7. Vùng nhận thức (Receptive field)	22
Hình 4.8. Dòng dữ liệu trong LeNet (Nguồn [25])	22
Hình 4.9. Khối phần dư sử dụng kết nối tắt (Residual block, Nguồn [31]).....	24
Hình 4.10. Khối dày đặc 5 lớp (5-layer dense block, Nguồn [32])	24
Hình 4.11. Minh họa về Image Segmentation (Nguồn [34])	25
Hình 4.12. Kiến trúc U-Net (Nguồn [36])	27
Hình 4.13. Kiến trúc trải phẳng của RNN (Nguồn [37])	28
Hình 4.14. Module lặp lại trong một RNN chứa một lớp duy nhất (Nguồn [37])....	28
Hình 4.15. One-hot encoding (Nguồn [38]).....	29
Hình 4.16. Một embedding 4 chiều (Nguồn [38])	29
Hình 4.17. Một khối LSTM với 2 nút ẩn và 1 đầu vào 3 chiều (Nguồn [41]).....	30
Hình 4.18. Tính toán trạng thái ẩn trong GRU (Nguồn [43])	31
Hình 4.19. Kiến trúc mô hình chuỗi sang chuỗi (Seq2Seq).....	32
Hình 4.20. Mô hình sinh từ thứ t từ câu đầu vào sử dụng Attention (Nguồn [47]) ..	34
Hình 4.21. Tính toán đầu ra của tầng tập trung (Nguồn [49])	34
Hình 4.22. Mô hình Seq2Seq sử dụng cơ chế Attention.....	35
Hình 4.23. Một từ sẽ chú ý tới các từ liên quan trong Self-Attention (Nguồn [50])	35
Hình 4.24. Dot-Product Attention (Nguồn [50]).....	36
Hình 4.25. Kiến trúc Transformer so với Attention-Seq2Seq (Nguồn [54])	37
Hình 5.1. Quy trình xây dựng bộ dữ liệu NomNaOCR	39
Hình 5.2. Toàn thư, Quyển thủ, tờ 1a trên website của VNPF	40
Hình 5.3. Cấu trúc thường thấy trong các tác phẩm thơ lục bát	41

Hình 5.4. Cấu trúc bất định trong Đại Việt Sử Kí Toàn Thư.....	41
Hình 5.5. Minh họa một cột có thể gồm nhiều bounding box	42
Hình 5.6. Cách gán nhãn với vị trí và số lượng ký tự cho Patch (màu đỏ).....	44
Hình 5.7. Kết quả đánh giá các annotator	46
Hình 5.8. Ảnh vết mực	48
Hình 5.9. Ảnh vết rách	48
Hình 5.10. Ảnh chữ mờ.....	48
Hình 5.11. Ảnh bị gấp đè	49
Hình 5.12. Ảnh trống	49
Hình 5.13. Ảnh bị dịch thiêu (vùng khoanh màu đỏ không có phần dịch).....	49
Hình 5.14. Ảnh bị dịch dư.....	50
Hình 5.15. Ảnh có ký tự khó để thấy	50
Hình 6.1. Xử lý ảnh căn bản thất bại khi các ký tự gần nhau (Nguồn [17]).....	56
Hình 6.2. Kiến trúc mô hình EAST.....	58
Hình 6.3. Kiến trúc của ResNet 18	59
Hình 6.4. Kiến trúc mô hình DBNet (Nguồn [56])	61
Hình 6.5. Các kiến trúc Injection và Merging trong Image Captioning	65
Hình 6.6. Mô hình Image Captioning dựa trên Attention	65
Hình 6.7. Bổ sung tọa độ pixel cho mô hình Attention-based Image Captioning	66
Hình 6.8. Mô hình CRNN (Nguồn [65]).....	67
Hình 6.9. Receptive field trong CRNN (Nguồn [65])......	68
Hình 6.10. Xác suất theo các bước thời gian của từ “sun” để tính toán CTC.....	70
Hình 6.11. Mô hình CNN kết hợp Attention-Seq2Seq	71
Hình 6.12. Mô hình CNN kết hợp kiến trúc Transformer	72
Hình 6.13. Mô hình CNN kết hợp chỉ Transformer Decoder	73
Hình 8.3. Ví dụ phát hiện sai trên tập thơ của mô hình DBNet	95
Hình 8.4. Ví dụ phát hiện đúng hoàn toàn của mô hình DBNet	95
Hình 8.5. Ví dụ phát hiện các văn bản nằm ngoài nội dung	96
Hình 8.6. Ví dụ phát hiện thiếu sót các văn bản	96

Hình 8.7. Ví dụ không phát hiện được ký tự “—” ở đầu patch	97
Hình 8.8. Phân phối các ký tự dự đoán sai của tập Validate.....	98
Hình 8.9. Ví dụ các dự đoán sai	98

DANH MỤC BẢNG

Bảng 5.1. Thống kê dữ liệu thô sau khi thu thập	42
Bảng 5.2. Thống kê các page lệch nhau giữa website của VNPF và annotator.....	47
Bảng 5.3. Số lượng câu theo độ dài	51
Bảng 5.4. Tần suất ở cấp độ ký tự của bộ dữ liệu NomNaOCR	51
Bảng 5.5. Thống kê giao nhau giữa 2 tập Train và Validate trong NomNaOCR	52
Bảng 5.6. Thống kê các Patch trong tập Train và Validate theo từng tác phẩm.....	53
Bảng 5.7. Thống kê các Page trong tập Train và Validate theo từng tác phẩm.....	54
Bảng 5.8. Thống kê số lượng câu theo độ dài trong Synthetic Nom String.	55
Bảng 5.9. Tần suất ở cấp độ ký tự của bộ dữ liệu Synthetic Nom String	55
Bảng 7.1. Cài đặt CNN backbone cho Text Recognition	76
Bảng 7.2. Cài đặt GRU units cho các mô hình Recognition.....	78
Bảng 7.3. Cài đặt cho các mô hình Recognition tận dụng kiến trúc Transformer....	78
Bảng 8.1. Kết quả tổng quan cho Text Detection	84
Bảng 8.2. Kết quả chi tiết của mô hình DBNet theo từng tác phẩm.....	84
Bảng 8.3. Kết quả chi tiết của mô hình EAST theo từng tác phẩm	85
Bảng 8.4. Kết quả Pre-training trên bộ Synthetic Nom String thuộc IHR-NomDB.	86
Bảng 8.5. Kết quả Fine-tuning và Retraining cho bộ dữ liệu NomNaOCR	88
Bảng 8.6. Kết quả các ngưỡng 10 khi Fine-tuning và Retraining cho NomNaOCR	90
Bảng 8.7. Kết quả End-to-End trên toàn bộ ảnh của tập Validate	92
Bảng 8.8. Kết quả End-to-End trên các ảnh thơ của tập Validate	93
Bảng 8.9. Kết quả End-to-End trên các ảnh văn xuôi của tập Validate.....	94
Bảng 8.10. Số ký tự vẫn được dự đoán đúng dù ít xuất hiện trong tập Train.....	98

DANH MỤC TỪ VIẾT TẮT

STT	Từ viết tắt	Ý nghĩa
1	BN	Batch Normalization
2	CER	Character Error Rate
3	CNN	Convolutional Neural Network
4	CRNN	Convolutional Recurrent Neural Network
5	CRW	Correctly Recognized Words
6	CTC	Connectionist Temporal Classification
7	char_acc	Character Accuracy
8	ĐVSKTT	Đại Việt Sử Ký Toàn Thư
9	FCN	Fully Convolutional Network
10	GRU	Gated Recurrent Unit
11	IoU	Intersection-over-Union
12	LSTM	Long Short-Term Memory
13	NMS	Non-maximum Suppression
14	OCR	Optical Character Recognition
15	PCC	Psedo-Character Center
16	RNN	Recurrent Neural Network
17	seq_acc	Sequence Accuracy
18	Seq2Seq	Sequence to Sequence
19	STR	Scene Text Recognition
20	TP	True Positive
21	VNPF	Vietnamese Nom Preservation Foundation

TÓM TẮT KHÓA LUẬN

Trong đề tài này, chúng tôi thực hiện xây dựng bộ dữ liệu NomNaOCR dành cho chữ Hán-Nôm dựa trên 3 tác phẩm lớn và có giá trị cao trong lịch sử Việt Nam gồm: Đại Việt Sử Ký Toàn Thư, Truyện Kiều, Lục Vân Tiên. Với 2953 trang sách viết tay được thu thập từ Hội Bảo tồn di sản chữ Nôm Việt Nam để phân tích và gán nhãn bán thủ công, từ đó tạo ra thêm 38318 bản cắt cho các vùng ảnh chứa văn bản cùng các chuỗi ký tự kỹ thuật số tương ứng, khiến đây trở thành bộ dữ liệu dành cho chữ Hán-Nôm lớn nhất Việt Nam hiện tại, phục vụ cho 2 bài toán trong nhận dạng ký tự quang học: phát hiện (Detection) và nhận dạng (Recognition) văn bản.

Một điểm đặc biệt là các triển khai của chúng tôi đều sẽ được thực hiện ở mức chuỗi, điều này chẳng những giúp tiết kiệm chi phí gán nhãn mà còn giúp chúng tôi giữ lại được ngữ nghĩa trong câu thay vì chỉ thực hiện cho từng ký tự riêng lẻ như các công trình trước. Từ đó, chúng tôi có được tiền đề để tập trung giải quyết 2 bài toán trên bằng nhiều phương pháp mới lạ. Với bài toán Detection, chúng tôi đề xuất 2 mô hình EAST và DBNet, tương ứng 2 hướng tiếp cận là Regression-based và Segmentation-based. Cùng với đó là 4 hướng giải quyết khác nhau cho bài toán Recognition gồm: Sinh mô tả cho ảnh, Mạng Nơ-ron Hồi tiếp Tích chập (CRNN), Mô hình Seq2Seq trong dịch máy, và tận dụng kiến trúc Transformer.

Thử nghiệm trên tập Validate của NomNaOCR cho thấy, DBNet vượt trội trong tác vụ phát hiện văn bản với F1-score lên tới 99.65%. Với riêng tác vụ nhận dạng, mô hình CNN kết hợp cùng kiến trúc Transformer sau khi được Fine-tuning và thêm vào một Kết nối tắt mà chúng tôi đề xuất đã đạt kết quả khá tốt với độ chính xác ở mức ký tự là 84.90% và một độ lỗi ký tự ở mức thấp với 13,35%. Ngoài ra, chúng tôi cũng thực hiện đánh giá End-to-End cho 2 bài toán trên và nhận được kết quả khá tốt khi sự kết hợp giữa DBNet và CRNN là vượt trội nhất với F1-score đạt 92.32%. Bên cạnh đó, nhiều phân tích chi tiết khác như kết quả phát hiện cho riêng 2 loại ảnh thơ và văn xuôi, kết quả nhận dạng trên các ngưỡng 10 ký tự hay các phân tích lỗi cho các mô hình tốt nhất cũng sẽ được chúng tôi đưa ra và làm rõ.

Chương 1. MỞ ĐẦU

1.1. Đặt vấn đề

Tiếng nói là khả năng bẩm sinh của con người, còn chữ viết là biểu thị cho nền văn minh của một đất nước, một phát minh sáng tạo của một dân tộc. Tiếng Việt diệu kì với ngũ âm cực kỳ phong phú cùng hệ thống chữ viết giàu mạnh nhất vùng Đông Á. Xuyên suốt chiều dài lịch sử, chữ viết nước ta đã trải qua hành trình từ chữ Hán hay chữ Nho đến chữ Nôm và cuối cùng là chữ Quốc Ngữ dựa trên hệ thống chữ Latin và đi cùng với mỗi loại chữ ấy là một trang sử vẻ vang đáng ôn lại của dân tộc.

Sau khi Ngô Quyền đánh tan quân Nam Hán trên sông Bạch Đằng năm 938, kết thúc nghìn năm Bắc thuộc, ông cha ta với ý thức tự chủ ngôn ngữ, đã sáng tạo ra chữ Nôm dựa trên cơ sở chữ Hán được đọc theo âm Hán-Việt, nên có thể nói chữ Hán là một tập con của chữ Nôm. Và trong hơn 1000 năm sau đó, từ thế kỷ 10 đến thế kỷ 20, song song với việc dùng chữ Hán, chữ Nôm được dùng để ghi lại phần lớn các tài liệu văn học, y học, triết học, tôn giáo, lịch sử văn hóa dân tộc. Tuy nhiên, di sản này hiện tại có nguy cơ tiêu vong bởi sự chuyển dịch sang loại chữ viết hiện đại hơn - chữ Quốc Ngữ.

Theo đó, dựa trên thông tin biết được từ Hội Bảo tồn di sản chữ Nôm Việt Nam (*Vietnamese Nôm Preservation Foundation - VNPF*) [1] thì: “Ngày nay, trên thế giới chưa có đến 100 người đọc được chữ Nôm. Một phần to tát của lịch sử Việt Nam như thế nằm ngoài tầm tay của 80 triệu người nói tiếng Việt”. Do giá trị to lớn của các tài liệu lịch sử đối với việc nghiên cứu, đặc biệt là các khía cạnh xã hội và lối sống thời trước cùng với những thông điệp mà cha ông để lại, việc bảo tồn di sản văn hóa này là cấp thiết. Các cơ quan của Việt Nam và các tổ chức trên thế giới đã sưu tầm hàng nghìn tập sách Hán-Nôm. Gần đây, nhờ sự phát triển của công nghệ thông tin trong việc bảo quản, quản lý, nghiên cứu và khai thác nguồn tài liệu Hán-Nôm, trên 4.000 văn bản đã được scan thành ảnh kỹ thuật số. Tuy nhiên, để sử dụng nguồn tri thức khổng lồ này, chúng phải được lập chỉ mục, dưới dạng văn bản tìm kiếm được đầy đủ, chú thích, trích dẫn và được dịch sang Quốc Ngữ hiện đại. Do

việc dịch thuật khó khăn và tốn nhiều thời gian cùng số lượng chuyên gia hạn chế nên các nỗ lực này không thể thực hiện trong một thời gian ngắn. Các kỹ thuật nhận dạng ký tự quang học sẽ tăng tốc quá trình số hóa này góp phần làm mọi công trình chính trong Hán-Nôm thành sẵn có trực tuyến.

1.2. Lý do chọn đề tài

*Dân ta phải biết sử ta
Cho tương gốc tích nước nhà Việt Nam*
(Trích “Việt Nam Quốc Sử Diễn Ca”, Hồ Chí Minh)

Lịch sử Việt Nam là lịch sử có chiều rộng, lại có chiều sâu, vì vậy Việt Nam không chỉ xanh hoa tốt lá mà còn mập gốc chắc rễ. Nền độc lập của cõi Việt đã được hoàn thành trong êm đềm của thời bình nên nước Việt Nam chẳng khác gì một quả chín rụng ra khỏi cây mẹ để tự sống một cuộc đời riêng, mang đầy đủ sinh lực trong chính mình. Khi một cây đã mang đầy đủ sinh lực trong chính mình và đã có gốc mập rễ sâu thì một cành có thể bị gãy và rạn nứt, thân cây có thể bị đốn nhưng cây không sao chết được. Từ gốc nó, người ta sẽ thấy mầm nảy lên và cây sống lại.

Với tình yêu cho những trang sử vẻ vang của dân tộc cùng khát khao cho cội nguồn hào hùng đó được tiếp tục duy trì và trở nên gần gũi hơn với từng người Việt, chúng tôi đã lựa chọn thực hiện đề tài này một cách đầy tâm huyết cùng với niềm tự hào trên từng dòng chữ viết được và đó cũng như là một cách chúng tôi tỏ lòng mình với công lao của cha ông ngày trước.

Ngoài ra, một lý do khác nữa cũng không kém cạnh chính là lượng kiến thức mà chúng tôi sẽ nhận được khi thực hiện đề tài này vì những gì chúng tôi sẽ làm cũng là đại diện cho một thách thức thực sự trong thực tế và đây đồng thời cũng là chủ đề hội tụ cho tất cả những gì chúng tôi đã học: từ nơi giao thoa giữa những kiến thức nền tảng trong chuyên ngành như các Kỹ thuật lập trình hay Thu thập dữ liệu và tiền xử lý dữ liệu, cho tới sự dung hòa giữa Thị giác máy tính và Xử lý ngôn ngữ tự nhiên. Nó gói gọn rất nhiều bài toán trong Khoa học dữ liệu cùng với sự mới lạ của nhiều phương pháp tiếp cận trong thế giới thực.

1.3. Mục tiêu khóa luận

Trong khóa luận này, chúng tôi tập trung nghiên cứu vào quy trình xây dựng một bộ dữ liệu tốt, các kỹ thuật xử lý ảnh, xử lý ngôn ngữ tự nhiên, và các phương pháp học sâu để giải quyết 2 bài toán: phát hiện và nhận dạng các ký tự Hán-Nôm viết tay trong các văn bản cũ. Vì thế, chúng tôi đã đặt ra các mục tiêu sau:

- Thứ nhất, tạo ra một bộ dữ liệu tốt với tên NomNaOCR, gồm 2953 trang (Page) là các ảnh quét của các văn bản cũ, cùng với 38318 bản cắt (Patch) cho các vùng ảnh chứa văn bản được trích xuất từ các trang này nhằm phục vụ cho các bài toán phát hiện và nhận dạng các ký tự Hán-Nôm viết tay. Bộ dữ liệu sẽ được cung cấp miễn phí nhằm mục đích nghiên cứu khoa học.
- Thứ hai, tiến hành cài đặt, triển khai thực nghiệm các phương pháp học sâu trên bộ dữ liệu NomNaOCR cho bài toán phát hiện văn bản (DBNet, EAST) và bài toán nhận dạng văn bản thông qua các hướng tiếp cận gồm: Sinh mô tả cho ảnh, Mạng Nơ-ron Hồi tiếp Tích chập, Mô hình chuỗi sang chuỗi trong dịch máy, và tận dụng kiến trúc Transformer. Sau đó, chúng tôi sẽ đánh giá và phân tích kết quả để tìm ra mô hình phù hợp cho từng bài toán. Ngoài ra, chúng tôi cũng sẽ thực hiện đánh giá kết hợp (End-to-End) cho 2 bài toán trên.

1.4. Đối tượng và phạm vi nghiên cứu

- **Đối tượng:** Bộ dữ liệu và các phương pháp học sâu để giải quyết các bài toán phát hiện và nhận dạng các ký tự Hán Nôm trong văn bản cổ.
- **Phạm vi:** Luận văn tập trung chủ yếu vào xây dựng bộ dữ liệu, cùng với phát hiện và nhận dạng văn bản. Cụ thể giới hạn trên bài toán phát hiện và nhận dạng các ký tự Hán Nôm trong các văn bản cũ.

Về giới hạn nghiên cứu, chúng tôi sẽ tập trung chủ yếu các vấn đề sau:

- Nghiên cứu quy trình xây dựng bộ dữ liệu, các phương pháp tiếp cận cho 2 bài toán phát hiện và nhận dạng văn bản.
- Thủ nghiệm và tìm ra các mô hình tốt cho 2 bài toán trên.

1.5. Các nội dung chính

Khóa luận này sẽ gồm 9 chương với các nội dung chính lần lượt như sau:

- **Chương 1. Mở đầu:** đặt ra các vấn đề, trình bày lý do thực hiện luận văn này để giải quyết các vấn đề được nêu. Tiếp đến là thiết lập các mục tiêu cần đạt được. Cuối cùng, giới thiệu sơ lược nội dung của từng chương trong luận văn.
- **Chương 2. Tổng quan:** giới thiệu về đề tài số hóa các văn bản cũ được viết tay bằng chữ Hán Nôm. Đặc biệt là tính ứng dụng thực tế của đề tài này trong việc lưu trữ tài liệu lịch sử.
- **Chương 3. Nghiên cứu liên quan:** giới thiệu các công trình nghiên cứu trong và ngoài nước liên quan đến quy trình xây dựng dữ liệu, các phương pháp được sử dụng để giải quyết bài toán phát hiện và nhận dạng chữ viết tay trong các văn bản cũ.
- **Chương 4. Cơ sở lý thuyết:** trình bày các kiến thức nền tảng mà chúng tôi áp dụng để xây dựng các phương pháp tiếp cận nhằm mục đích giải quyết các bài toán đã đặt ra trong đề tài này.
- **Chương 5. Bộ dữ liệu NomNaOCR:** trình bày quy trình xây dựng bộ dữ liệu đạt chuẩn và có chất lượng tốt cũng như thực hiện thống kê chi tiết và phân tích các đặc điểm của dữ liệu, từ đó làm tiền đề để phát triển các phương pháp cho bài toán đã đặt ra trên bộ dữ liệu này.
- **Chương 6. Các phương pháp tiếp cận:** trình bày các hướng tiếp cận mà chúng tôi đã nghiên cứu và áp dụng trên bộ dữ liệu NomNaOCR cho bài toán phát hiện và nhận dạng các ký tự Hán Nôm viết tay trong các văn bản cũ.
- **Chương 7. Cài đặt thử nghiệm:** trình bày các bước thiết lập cùng cài đặt chi tiết cho các siêu tham số và thiết bị dùng để huấn luyện các mô hình.
- **Chương 8. Đánh giá và kết quả:** trình bày các kết quả mà chúng tôi đã thu được đồng thời thực hiện đánh giá, giải thích các kết quả đạt được đó và phân tích các lỗi của mô hình tốt nhất khi dự đoán trên bộ dữ liệu NomNaOCR.
- **Chương 9. Kết luận và hướng phát triển:** tổng kết các thành quả đã đạt được và đề xuất các phương pháp trong tương lai để cải thiện hiệu suất của mô hình.

Chương 2. TỔNG QUAN

2.1. Giới thiệu về tài

Sự cần thiết của việc số hóa đang tăng lên nhanh chóng trong thời kỳ hiện đại. Do sự phát triển của thông tin, các công nghệ kết nối và sự sẵn có rộng rãi của các thiết bị cầm tay, mọi người thường thích nội dung số hóa hơn các tài liệu in, bao gồm cả sách và báo. Ngoài ra, việc tổ chức dữ liệu số hóa và phân tích chúng cho các mục đích khác nhau cũng trở nên dễ dàng hơn nhờ các kỹ thuật tiên tiến như trí tuệ nhân tạo, ... Vì vậy, để theo kịp với bối cảnh công nghệ hiện nay, cần phải chuyển đổi tất cả các thông tin hiện tại từ định dạng in sang định dạng số hóa.

Thật vậy, việc số hóa các tài liệu ở định dạng giấy là vô cùng quan trọng, đặc biệt là các tài liệu lịch sử có ý nghĩa rất lớn đối với việc bảo tồn di sản văn hóa. Vì thế, nhận thức được tầm quan trọng và tính cấp bách của việc số hóa các tài liệu lịch sử trên thế giới đã và đang nhận được sự quan tâm và thu hút của nhiều nhà nghiên cứu. Tại Hàn Quốc, Kim và cộng sự đã phát triển một hệ thống số hóa hơn 10 triệu tài liệu Hanja viết tay - một loại chữ viết phổ biến ở Hàn Quốc cho đến cuối thế kỷ thứ 9 [2]. Tại Trung Quốc, công ty Digital Heritage Publishing Ltd. đã số hóa hơn 36.000 tập tương đương 4,7 triệu trang của Tú khố toàn thư - bộ sách lớn nhất do 361 học giả thời Càn Long biên soạn [3].

Tại Việt Nam, từ thế kỷ 10 đến thế kỷ 20, một số lượng rất lớn các tài liệu đều được ghi chép bằng chữ Nôm - một hệ thống chữ viết cũ của Việt Nam và vẫn đang được lưu trữ trong các đình, chùa, và thư viện. Hiện tại, các tác phẩm lịch sử vô giá này đang có nguy cơ bị mai một, khó để tiếp cận đến được các thế hệ sau và đang dần bị xuống cấp, hầu hết chưa được số hóa. Bên cạnh đó, số học giả hiểu biết và đọc được các văn bản này ngày càng ít dần. Vì vậy, tại Việt Nam cũng đã có một số công trình nghiên cứu như [4] hay [5] nhằm mục đích giải quyết các vấn đề trên nhưng vẫn còn nhiều hạn chế. Đặc biệt là chưa có công trình nào cung cấp được một bộ dữ liệu đủ tốt để phục vụ cho cả 2 bài toán phát hiện và nhận dạng các ký tự Hán Nôm viết tay nhằm số hóa các tài liệu lịch sử ở Việt Nam.

Bài toán phát hiện và nhận dạng các ký tự Hán Nôm trong các văn bản lịch sử cũ là một nhiệm vụ đầy thách thức, có sự thu hút với giới học thuật. Trong khóa luận này, chúng tôi sẽ tập trung giải quyết cả 2 nhiệm vụ trên. Như vậy, với một trường hợp cụ thể là một trang giấy viết tay bằng chữ Hán Nôm, các nhiệm vụ sẽ được định nghĩa như sau:

- **Đầu vào:** Ảnh quét của trang văn bản viết tay bằng các ký tự Hán Nôm.
- **Đầu ra:** Các chuỗi ký tự Hán-Nôm tương ứng có trong ảnh đầu vào đó dưới dạng kỹ thuật số.

2.2. Tính ứng dụng của đề tài

Các tài liệu lịch sử ghi lại nhiều thăng trầm và biến động hay cả những bài học quý giá của cha ông trải dài trong quá khứ đều được chép lại bằng chữ Hán Nôm. Vì vậy, việc số hóa các văn bản cũ đó còn là trách nhiệm, góp phần bảo tồn và gìn giữ những giá trị nước nhà. Nhưng hiện nay, các văn bản này vẫn còn đang được lưu trữ chủ yếu ở định dạng giấy, và đang ngày càng xuống cấp. Để số hóa được chúng, cần các tổ chức bảo tồn thực hiện nhập liệu một cách thủ công, nhưng sức người thì lại có hạn và số lượng người có thể hiểu để mà viết được thì lại càng hạn chế. Dẫn đến chi phí cho việc bảo tồn các văn bản này là rất cao hay xa hơn nữa là khiến sự tiếp cận của những người yêu thích chữ Hán-Nôm nói riêng và lịch sử Việt Nam nói chung sẽ bị hạn chế, kéo theo việc những người hiểu biết về loại chữ này đã ít nay còn ít hơn.

Vì vậy, cần có một công cụ có thể trợ giúp các tổ chức chuyển đổi từ việc số hóa thủ công sang số hóa một cách tự động, từ đó giúp tiết kiệm được phần lớn thời gian và công sức. Do đó, tính ứng dụng của đề tài này là cấp thiết, nên chúng tôi tập trung vào cả 2 nhiệm vụ phát hiện và nhận dạng các ký tự Hán Nôm viết tay, có thể hiểu đơn giản là khi ảnh quét của một trang văn bản cũ chứa các ký tự Hán Nôm được đưa ra, phương pháp của chúng tôi sẽ chỉ ra các vị trí cụ thể của các văn bản trong hình và trích xuất các ký tự Hán Nôm trong đó một cách tự động dưới dạng kỹ thuật số.

2.3. Thách thức

Nhìn chung, các bài toán về phát hiện và nhận dạng các ký tự Hán Nôm có rất nhiều vấn đề khó khăn và thách thức. Trong phần phát hiện, các vùng ảnh có văn bản thì các đối tượng cần phát hiện là các ký tự có kích thước nhỏ dẫn đến khó phát hiện hơn các bài toán phát hiện đối tượng đơn thuần. Các trang sách cũ viết bằng hệ thống chữ viết Hán Nôm có cấu trúc các câu được viết theo chiều dọc nhưng trong một cột có thể 2 hai cột nhỏ khác bên trong và nằm rất sát nhau dẫn đến khá khó khăn trong việc phát hiện ranh giới giữa các vùng ảnh chứa văn bản. Khác với bài toán phát hiện đối tượng thông thường, mật độ của các đối tượng trong ảnh khá thưa nhưng đối với các ảnh trong đề tài này thì có mật độ văn bản khá dày đặc, đây cũng là một thách thức không nhỏ cho việc phát hiện.

Với bài toán nhận dạng các ký tự Hán Nôm, do các ảnh đã khá cũ nên các ký tự cũng bị phai mực theo thời gian, chất lượng giấy xuống cấp và ô vàng, bị rách hay bị lem mực cũng là những vấn đề nan giải cho việc nhận dạng. Bên cạnh đó, giống với các hệ thống chữ viết khác của các nước như Trung Quốc, Hàn, Nhật thì chữ Nôm cũng là chữ tượng hình vì thế các nét trong mỗi ký tự là vô cùng quan trọng, việc thiếu dù chỉ một nét nhỏ cũng có thể dẫn đến một nghĩa khác hoặc thậm chí không tồn tại.

Cuối cùng, để tạo ra được một bộ dữ liệu chuẩn và tốt là không hề đơn giản, đặc biệt là bộ dữ liệu dành cho cả 2 bài toán nêu trên, vì các thành viên trong nhóm không được học về chữ Hán Nôm nên không thể đánh máy cũng như là viết được các chữ này. Bên cạnh đó, chi phí cho việc gán nhãn trực tiếp các ký tự vào các vùng ảnh chứa văn bản là quá lớn và tốn nhiều thời gian.

Chương 3. NGHIÊN CỨU LIÊN QUAN

3.1. Tình hình nghiên cứu trên thế giới

Những năm gần đây, công việc chuyển đổi số đang rất được quan tâm trên toàn thế giới. Trong đó, công cuộc số hóa văn bản giấy đang được đẩy mạnh và phát triển không ngừng. Cùng với sự bức phá về các kỹ thuật học máy và học sâu, việc số hóa văn bản đã và đang đạt được những thành tựu nhất định, góp phần đẩy mạnh nhanh chóng công việc chuyển đổi số. Vì vậy, 2 bài toán phát hiện (Detection) và nhận dạng (Recognition) văn bản đang thu hút rất lớn các nhà nghiên cứu và doanh nghiệp. Tuy nhiên, sự kết hợp cho cả 2 bài toán này là một nhiệm vụ đầy thách thức và đóng vai trò quan trọng trong nhiều lĩnh vực. Vì thế, nhiều công trình liên quan đến chủ đề này đã được công bố.

Đối với riêng các hình ảnh tài liệu lịch sử, việc số hóa có ý nghĩa rất quan trọng đối cho mục đích bảo tồn di sản văn hóa. Hơn nữa, việc thu được các ký tự kỹ thuật số từ hình ảnh văn bản trong quá trình số hóa là cần thiết để cung cấp khả năng truy cập thông tin hiệu quả vào nội dung của các tài liệu này. Nhận dạng văn bản viết tay đã trở thành một chủ đề nghiên cứu quan trọng trong các lĩnh vực xử lý hình ảnh và ngôn ngữ. Vì thế với việc xây dựng các bộ dữ liệu và phát triển các mô hình, trên thế giới cũng đã có một số nghiên cứu tiêu biểu cho chủ đề này.

Theo đó, tác giả A.D. Le cùng các cộng sự đã giới thiệu mô hình Attention-based Encoder-Decoder để nhận dạng các tài liệu lịch sử của Nhật Bản [6] sử dụng 2 mô-đun chính: DenseNet để trích xuất các đặc trưng và bộ giải mã LSTM được tích hợp để tạo văn bản đầu ra. Mô hình yêu cầu hình ảnh đầu vào là dòng văn bản để cho ra các ký tự đầu ra tương ứng. Do đó, nhóm tác giả không cần gán nhãn cho các ký tự nên tiết kiệm rất nhiều thời gian. Ngoài ra nhóm tác giả cũng giới thiệu một phương pháp tạo dòng văn bản nhân tạo để giải quyết vấn đề mất cân bằng của bộ dữ liệu. Mô hình đã đạt được tỷ lệ lỗi ký tự là 23,76% và 22,52% tương ứng với có và không huấn luyện với dòng văn bản nhân tạo. Hơn nữa, hệ thống nhận diện này vượt trội so với mô hình CNN-LSTM.

Mặt khác, E. Granell và các cộng sự đã đề xuất một giải pháp [7] với việc kết hợp một hệ thống nhận dạng quang học mạnh mẽ để đối phó với các hình ảnh bị nhiễu bằng mô hình ngôn ngữ dựa trên các đơn vị từ vựng phụ (sub-lexical) mà sẽ mô hình hóa các từ nằm ngoài bộ từ vựng (*Out of Vocabulary – OOV*). Cách tiếp cận mô hình hóa ngôn ngữ như vậy làm giảm kích thước của từ vựng trong khi tăng độ bao phủ. Các thử nghiệm đầu tiên được tiến hành trên tập dữ liệu Rodrigo có sẵn, chứa các văn bản số hóa của một bản thảo tiếng Tây Ban Nha cũ, với một công cụ nhận dạng dựa trên mô hình Markov ẩn (*Hidden Markov Model – HMM*). Họ chỉ ra rằng các đơn vị từ vựng phụ tốt hơn các đơn vị từ về tỷ lệ lỗi của từ (*Word Error Rate – WER*), tỷ lệ lỗi của ký tự (*Character Error Rate – CER*) và tỷ lệ accuracy của từ nằm trong OOV. Sau đó, cách tiếp cận này được áp dụng cho các bộ phân loại mạng sâu, cụ thể là Bi-LSTM và Mạng Hồi tiếp Tích Chập (*Convolutional Recurrent Neural Network – CRNN*). Kết quả thu được cho thấy CRNN hoạt động tốt hơn HMM và Bi-LSTM, đạt WER và CER thấp nhất cho tập dữ liệu hình ảnh này và cải thiện đáng kể khả năng nhận dạng OOV.

E. Chammas và các cộng sự cũng đã đưa ra các nhận định về các tài liệu lịch sử có nhiều thách thức đối với các hệ thống nhận dạng chữ viết tay (handwritten) [8]. Trong đó phải kể đến các bước phân vùng và ghi nhãn. Các dòng văn bản được chú thích (annotate) cẩn thận là cần thiết để huấn luyện một mô hình Deep Learning. Trong công trình này, nhóm tác giả đã trình bày cách huấn luyện một hệ thống nhận dạng văn bản với ít dữ liệu được gán nhãn. Cụ thể, họ huấn luyện một mô hình Convolutional Recurrent Neural Network (CRNN) chỉ trên 10% dòng văn bản được gán nhãn thủ công từ tập dữ liệu. Sau đó, đề xuất quy trình huấn luyện tăng cường bao gồm phần còn lại của dữ liệu. Hiệu suất được tăng lên bằng cách tăng tập huấn luyện với dữ liệu đa quy mô được chế tạo đặc biệt. Ngoài ra, nhóm tác giả cũng đề xuất một sơ đồ chuẩn hóa dựa trên mô hình mà xem xét sự thay đổi về tỷ lệ của các từ viết tay tại giai đoạn nhận dạng. Hệ thống này cũng đã đạt được kết quả tốt thứ hai trong cuộc thi ICDAR 2017 về nhận dạng các văn bản trong ảnh ngoại cảnh (Scene Text).

3.2. Tình hình nghiên cứu trong nước

Tại Việt Nam, để giải quyết những vấn đề cho chữ Nôm như đã đề cập trong 1.1 hay trong 2.1, nhiều nhà nghiên cứu trong nước cũng góp không ít công sức thông qua các công trình liên quan nhằm góp phần số hóa cho các tài liệu cũ được viết bằng chữ Nôm này.

Nổi bật nhất gần đây, M.T. Vu và các cộng sự đã giới thiệu bộ dữ liệu IHR-NomDB [9] dành cho hệ thống chữ viết cũ của Việt Nam. Hơn 260 trang của các tác phẩm chữ Nôm đã được thu thập từ Hội Bảo tồn di sản chữ Nôm Việt Nam để phân tích và gán nhãn, tác giả đã xác định các bounding box theo cách thủ công để tạo ra hơn 5000 bản cắt cho các ảnh chữ viết tay cùng các chữ Nôm ký thuật số tương ứng, và bản dịch sang chữ Quốc Ngữ. Bên cạnh bộ dữ liệu viết tay này, tác giả cũng đã tạo thêm bộ dữ liệu Synthetic Nom String gồm 101,621 hình ảnh được sinh tự động bằng cách sử dụng ngân hàng câu chữ Nôm đã được thu thập. Đây đã trở thành một cơ sở dữ liệu công khai đầu tiên và lớn nhất dành cho việc nghiên cứu chữ viết tiếng Việt cũ. Đối với các kết quả cơ bản, tác giả đã thực hiện kiểm tra trên tập xác thực của bộ dữ liệu chữ viết tay bằng cách sử dụng mô hình CRNN kết hợp CTC Loss được huấn luyện trước (Pre-training) trên bộ dữ liệu Synthetic Nom String và đạt được độ chính xác 42,70% ở cấp độ chuỗi và 82,28% cho cấp độ ký tự.

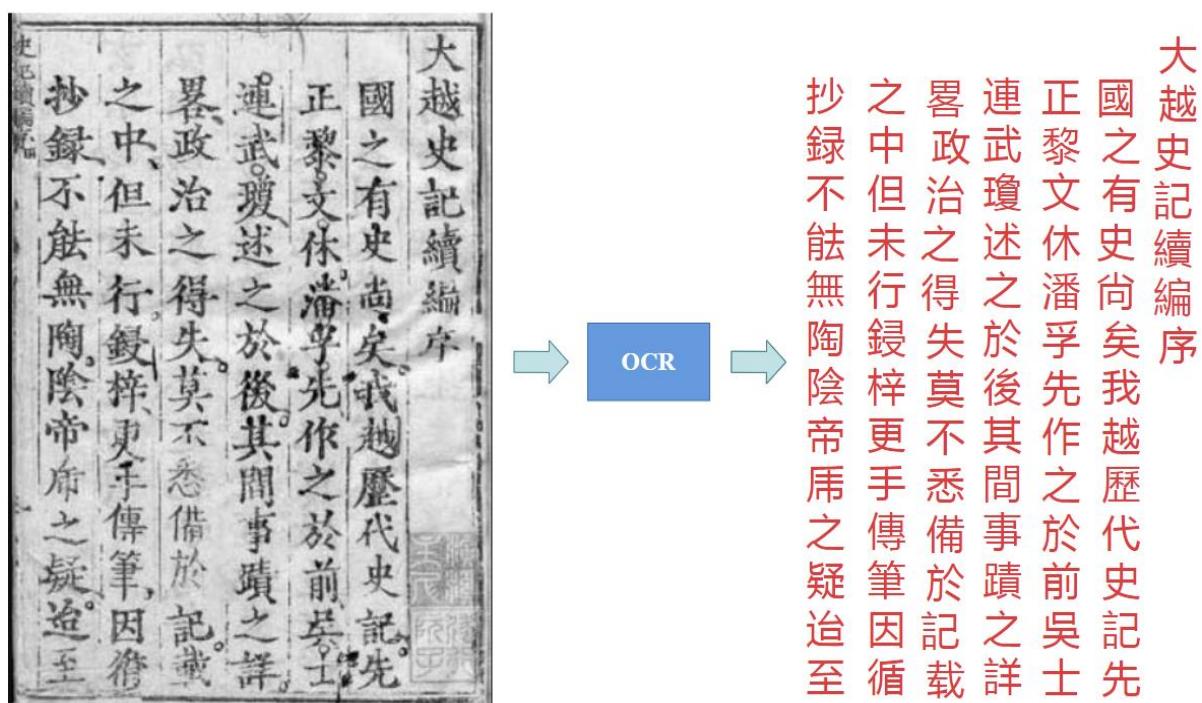
Ngoài ra, một phương pháp khác dựa trên phân vùng văn bản [10] để số hóa các tài liệu Nôm bằng cách sử dụng các mạng CNN sâu đã được đề xuất bởi K.C. Nguyen cùng các cộng sự. Các trang chữ Nôm được tiền xử lý, phân vùng thành các ký tự riêng biệt và được nhận dạng sau đó. Kiến trúc U-Net được áp dụng ở đây để tạo bản đồ phân vùng và trích xuất vùng ảnh chứa ký tự. Tiếp theo, tác giả cũng đề xuất các bộ phân loại để nhận dạng từng ký tự bằng việc sử dụng mô hình ngôn ngữ. So với phương pháp phân vùng truyền thống chỉ đạt được IoU là 81,23%, thì phương pháp sử dụng mạng CNN sâu tạo ra kết quả tốt hơn với IoU là 92,08% trong việc phát hiện các vùng ảnh chứa ký tự. Về nhận dạng ký tự, các mô hình CNN tác giả đề xuất tốt hơn các mô hình truyền thống với tỷ lệ nhận dạng là 85,07%.

K.C. Nguyen và các cộng sự cũng đã có đề xuất phương pháp nhận dạng ký tự khác trên một bộ dữ liệu lớn gồm 32.695 chữ Nôm [11]. Cho đến thời điểm bài báo được công bố, bộ dữ liệu lớn nhất mà các phương pháp nhận dạng ký tự đã được nghiên cứu là khoảng 10.000 điểm dữ liệu gồm tiếng Trung, tiếng Nhật và tiếng Hàn nhưng các ngôn ngữ cũ có nguồn gốc từ Trung Quốc chiếm đa số. Tác giả đã đề xuất một phương pháp nhận dạng một tập rất lớn các danh mục Nôm sử dụng các mạng CNN sâu. Phương pháp đề xuất này đã đưa ra các danh mục thô (coarse category) được chuẩn bị trước bởi K-means. Tác giả xây dựng các mạng CNN sâu gồm trình trích xuất đặc trưng danh mục thô, trình phân loại danh mục thô và trình phân loại danh mục được tinh chỉnh. Trước tiên nhóm tác giả đã thực hiện tiền huấn luyện trình trích xuất đặc trưng và trình phân loại với các danh mục thô. Sau đó đóng băng chúng và thực hiện tinh chỉnh (Fine-tuning) để nhận dạng các ký tự trong toàn bộ danh mục Nôm, không như các phân loại thô và phân loại được tinh chỉnh thông thường, mà sẽ được thực hiện song song với các feature map được tạo trong quá trình trích xuất đặc trưng. Thử nghiệm cho thấy rằng kiến trúc này cung cấp tỷ lệ nhận dạng tốt hơn so với các cách làm sử dụng GLVQ và MQDF trước đây với accuracy đạt 95.88% trong 30 epoch.

Chương 4. CƠ SỞ LÝ THUYẾT

4.1. Nhận dạng ký tự quang học (OCR)

Nhận dạng ký tự quang học (*Optical Character Recognition – OCR*) để cập đến một tập hợp các vấn đề về thị giác máy tính với mục đích chuyển đổi các hình ảnh kỹ thuật số hoặc hình ảnh tài liệu được scan thành dưới dạng văn bản mà máy tính có thể xử lý, lưu trữ và chỉnh sửa được như một tập tin văn bản thông thường hoặc dưới dạng một phần của phần mềm nhập liệu. Các hình ảnh có thể bao gồm tài liệu, hóa đơn, biểu mẫu pháp lý, chứng minh thư hoặc những thứ trong môi trường tự nhiên (OCR in the wild) như biển báo đường phố, biển số xe, ... [12]



Hình 4.1. OCR cho văn bản Hán-Nôm

Dù thường không được chú ý, nhưng OCR là một trợ giúp không thể thay thế khi ta nói về tự động hóa. Nó giúp loại bỏ các quy trình không cần thiết của các tài liệu giấy, cho phép ta phân loại, sắp xếp, lưu trữ, quản lý và chia sẻ thông tin, đồng thời tránh các rủi ro bảo mật liên quan đến bản chất vật lý của các tài liệu giấy. Ngoài ra, tự động hóa dựa trên OCR không chỉ là chia sẻ thông tin dưới dạng kỹ thuật số. Khi có nhiều tài liệu, các loại máy móc có thể sử dụng chúng làm mục nhập dữ liệu để

tìm kiếm các mẫu và xu hướng. Việc trực quan hóa cũng trở nên dễ dàng hơn: nếu ta cần một biểu đồ, lược đồ hoặc bảng tính, thì việc sử dụng các tài liệu kỹ thuật số sẽ nhanh hơn nhiều so với việc biên soạn một báo cáo trực quan bằng tay. OCR cho phép ta dành ít thời gian hơn để xử lý từng tài liệu mới, tiết kiệm chi phí nhân lực và thay vào đó tập trung vào các chiến lược gia tăng giá trị [13]. Đây là một chủ đề phức tạp vì nó là sự dụng hòa giữa 2 lĩnh vực lớn trong AI:

- Thị giác máy tính (*Computer Vision – CV*): huấn luyện các mô hình học máy để nhìn và giải thích hình ảnh theo cách tương tự như cách con người thực hiện.
- Xử lý ngôn ngữ tự nhiên (*Natural Language Processing – NLP*): chủ yếu xử lý văn bản hay các dữ liệu chuyển giọng nói thành văn bản và tập trung vào việc dạy cho máy tính hiểu lời nói của con người.

4.1.1. Các khái niệm cơ bản

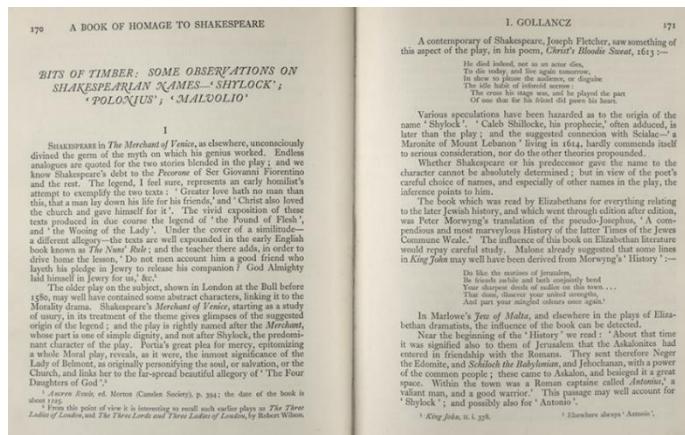
Thường có sự pha trộn qua lại giữa 3 khái niệm: nhận dạng ký tự quang học (*Optical Character Recognition – OCR*), nhận dạng ký tự thông minh (*Intelligent Character Recognition – ICR*) và thu thập dữ liệu thông minh (*Intelligent Data Capture – IDC*). Một số nguồn có xu hướng sử dụng các thuật ngữ này như các từ đồng nghĩa, nhưng có một sự khác biệt khá rõ ràng giữa chúng.

Thuật ngữ OCR thông thường được sử dụng phổ biến nhất cho văn bản trong các tài liệu in tiêu chuẩn, có cấu trúc. ICR là một phiên bản tiên tiến hơn của OCR được sử dụng cho các văn bản viết tay (*handwritten*), do sự phát triển của các công nghệ OCR hiện đại, các nhà khoa học dữ liệu và các kỹ sư hiếm khi phân biệt 2 hình thức thu thập dữ liệu tự động này. IDC đại diện cho các thuật toán được xây dựng để tự động hóa tốt hơn OCR vào các quy trình kinh doanh, nó kết hợp khả năng nhận dạng của OCR với việc giải thích dữ liệu (*data interpretation*) cho phép phân loại tài liệu và cả điểm nhập liệu [14]. Hiện tại, OCR không chỉ giới hạn ở nhận dạng văn bản tài liệu hoặc sách mà còn bao gồm các hình ảnh chứa văn bản được chụp trong các môi trường hỗn tạp hay không đồng nhất, hình thành nên các vấn đề như nền phức tạp, nhiễu, ánh sáng, font chữ khác nhau và biến dạng hình học trong ảnh, ...

4.1.2. Phân loại hình ảnh chúa văn bản

Các thách thức trong OCR phát sinh chủ yếu do các tác vụ OCR đang thực hiện. Đã có một số bài viết phân loại các hình ảnh chúa văn bản này thành 2 loại (Có và Phi cấu trúc) nhưng có vẻ cách phân loại này vẫn còn khá nhập nhằng và chưa được rõ ràng với loại dữ liệu của đề tài này nên chúng tôi sẽ chia chúng thành 3 loại sau:

- **Có cấu trúc (Structured):** văn bản trong các tài liệu được đánh máy, có nền sạch, đồng nhất, font chữ theo tiêu chuẩn, ít nhiễu (Noise), hàng lối rõ ràng, có một trật tự nhất định và thường có mật độ văn bản (Density) cao.



Hình 4.2. Ảnh chúa văn bản có cấu trúc (Nguồn [15])

- **Phi cấu trúc (Unstructured):** đây là tác vụ OCR nhiều khó khăn nhất, văn bản sẽ ở những vị trí ngẫu nhiên trong ảnh hay trong một hoạt cảnh tự nhiên (OCR in the wild), thường có mật độ văn bản thưa thớt, cấu trúc hàng lối bất định, nền phức tạp, và không có font chuẩn. Dữ liệu kiểu này cũng có thể được chia thành 3 loại nhỏ khác:
 - **Graphic text:** ảnh mà có văn bản được thêm vào sau khi đã có ảnh như phụ đề của video, ...
 - **Scene text:** ảnh mà có văn bản xuất hiện trong tự nhiên, tức văn bản là thành phần có sẵn trong ảnh. Các dữ liệu loại này có nhiều thách thức như về hướng (Orientation), loại font, điều kiện ánh sáng hay độ lệch (Skewness) của chữ. Tác vụ OCR cho dữ liệu loại này còn gọi là Scene Text Recognition (STR).



Graphic Text

Scene Text

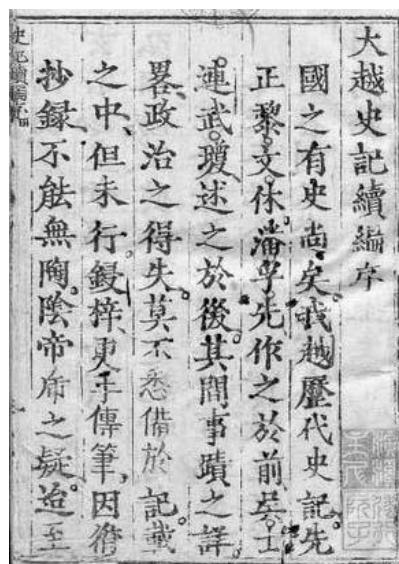
Hình 4.3. Ảnh chứa Graphic text và Scene text (Nguồn [16])

- Synthetic text: khá giống với Graphic text, đây là một ý tưởng khá hay để cải thiện hiệu suất của quá trình huấn luyện bằng cách tạo dữ liệu bằng máy tính. Việc sinh các ký tự hoặc từ ngẫu nhiên lên một hình ảnh sẽ có vẻ tự nhiên hơn nhiều so với bất kỳ vật thể nào khác vì tính chất phẳng của văn bản. Các bộ dataset cho dữ liệu loại này cũng vượt trội về khả năng tạo ra các ngôn ngữ khác nhau, ngay cả những ngôn ngữ khó, chẳng hạn như tiếng Trung, tiếng Do Thái, tiếng Ả Rập, ... hay cho chữ Hán-Nôm như trong bộ dữ liệu IHR-NomDB [9] được dùng để huấn luyện các mô hình pretrain cho đề tài này.



Hình 4.4. Ảnh chứa Synthetic text (Nguồn [17])

- Lai giữa có và phi cấu trúc (Hybrid): thường là các bản scan, photocopy không phải ở dạng đánh máy của các tài liệu, sách, có cấu trúc tốt, nền ít phức tạp, gọn gàng, thường có mật độ văn bản cao hơn so với các ảnh chứa văn bản phi cấu trúc nhưng có hướng nghiêng và độ lệch nhỏ. Ngoài ra, đối với loại văn bản như các tư liệu lịch sử thì còn có thể có sự những thay đổi lớn và mơ hồ do sự khác nhau về nét chữ giữa từng người hay chất lượng ảnh có thể bị giảm dần theo thời gian như ảnh của các tác phẩm lịch sử dùng trong bộ dữ liệu NomNaOCR chúng tôi đã xây dựng.



Hình 4.5. Ảnh chứa văn bản lai giữa có và phi cấu trúc trong NomNaOCR

4.1.3. OCR và Học sâu

OCR là một trong những nhiệm vụ thị giác máy tính được giải quyết sớm nhất vì ở khía cạnh nào đó, nó không cần tới học sâu (Deep Learning), do đó đã có những cách triển khai OCR khác nhau ngay cả trước khi Deep Learning bùng nổ vào năm 2012. Điều này khiến nhiều người nghĩ rằng các bài toán OCR đã được “giải”, nó không còn là một thách thức nữa và việc sử dụng Deep Learning cho OCR là một việc làm quá mức cần thiết [17].

Trên thực tế, OCR chỉ mang lại kết quả tốt trong các trường hợp cụ thể và đúng là có những giải pháp tốt cho một số tác vụ OCR không yêu cầu Deep Learning. Rất nhiều kỹ thuật trước đó đã giải quyết vấn đề OCR cho văn bản có cấu trúc bằng một

số kỹ thuật thị giác máy tính truyền thống hay các phương pháp xử lý ảnh căn bản như bộ lọc hình ảnh, hình thái học (morphology) và phát hiện cạnh (contour) để từ đó phân loại một vùng ảnh là chữ gì, ... Các kỹ thuật này chỉ hoạt động tốt trên các tập dữ liệu hẹp, theo khuôn mẫu (template-based), không thay đổi nhiều về hướng, vị trí văn bản hay chất lượng hình ảnh, ... nhưng chúng không hoạt động đúng với các môi trường ảnh không bị ràng buộc khác, có mật độ văn bản nhỏ hơn hay có các thuộc tính khác với dữ liệu có cấu trúc, ...

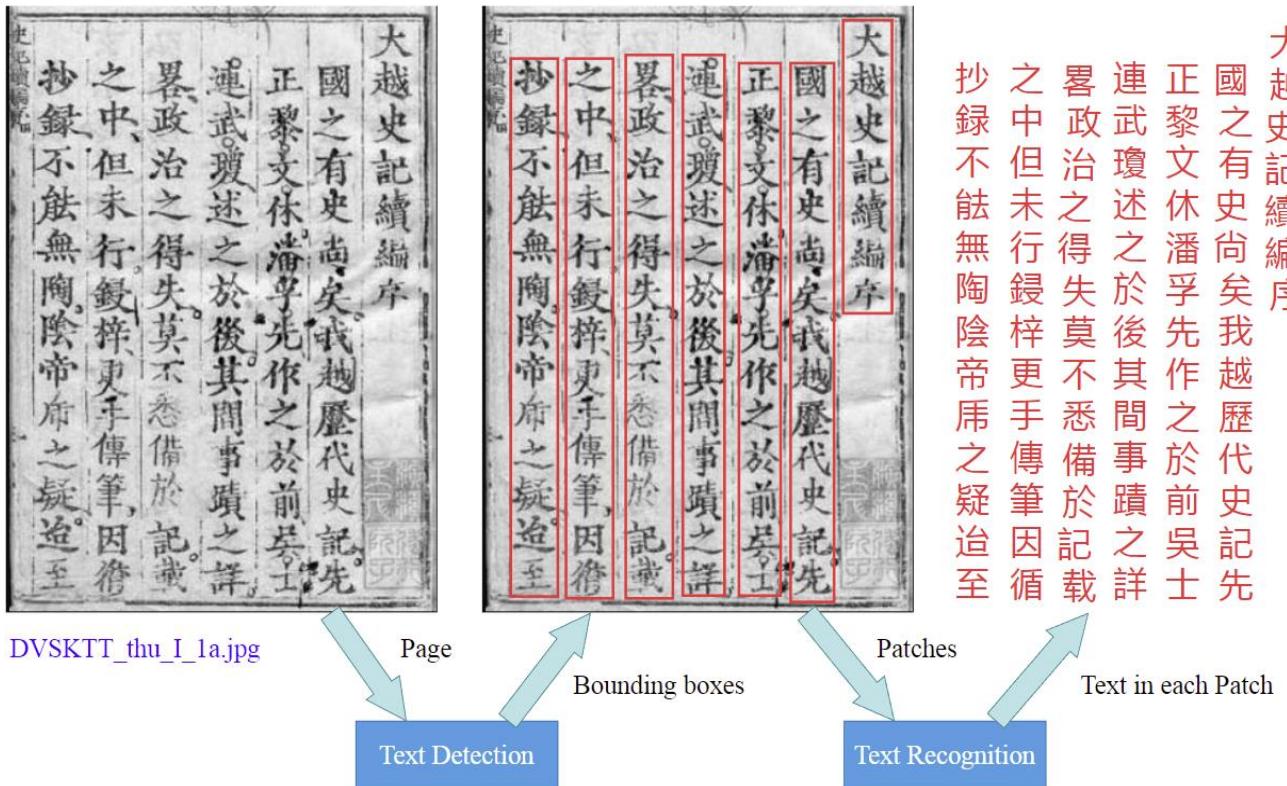
Nhìn chung, đây vẫn được xem là thách thức, để làm cho các mô hình trở nên mạnh mẽ với các biến thể trên, giúp các doanh nghiệp có thể triển khai các ứng dụng học máy của họ trên quy mô lớn, các giải pháp mới cần được đưa ra. Các phương pháp tiếp cận bằng Deep Learning đã được cải thiện trong vài năm qua, làm hồi sinh mối quan tâm đến vấn đề OCR, nơi mạng nơ-ron có thể kết hợp các tác vụ xác định văn bản trong một hình ảnh và hiểu văn bản đó là gì. Deep Learning có thể được xem là bắt buộc để tiến tới các giải pháp tốt hơn và tổng quát hơn [12].

4.1.4. Các bước triển khai chính

Có 2 mức độ để triển khai các tác vụ OCR: mức chuỗi (sequence level) và mức ký tự (character level). Ngày nay, mức chuỗi thường được ưa chuộng hơn do mức ký tự đòi hỏi chi phí gán nhãn quá lớn vì cần phải vẽ bounding box cho từng ký tự, từ đó dễ xuất hiện nhiều vấn đề nan giải hơn như bounding box giữa 2 ký tự liên tiếp có thể không rõ ràng hoặc bị đè lên nhau (overlap), ... Tiếp theo, người ta cũng chia OCR thành 2 bài toán con chính:

- Phát hiện văn bản (Text Detection): phát hiện vùng ảnh có chứa văn bản. Đầu vào là ảnh (hay Page đối với đề tài này), đầu ra là các khung chứa (bounding box) bao quanh các phần văn bản được tìm thấy trên ảnh.
- Nhận dạng văn bản (Text Recognition): sau khi detect được các bounding box hay các vùng ảnh có chứa văn bản, ta tách riêng từng vùng ảnh này ra từ ảnh gốc, tạo thành các phần ảnh nhỏ còn gọi là các Patch. Đầu vào lúc này sẽ là Patch và đầu ra là văn bản có trong Patch đó.

國之有史尚矣我越歷代史記先正黎文休潘孚先作之於前吳士連武瓊述之於後其間事蹟之中但未行鋟梓更手傳筆因循暑政治之得失莫不悉備於記載抄錄不能無陶陰帝廟之疑迨至



Hình 4.6. Pipeline với 2 bài toán con chính trong OCR

4.1.5. Một số dataset cho văn bản phi cấu trúc

Có rất nhiều bộ dữ liệu hình ảnh có sẵn cho các văn bản phi cấu trúc và cho tiếng Anh, nhưng sẽ khó hơn để tìm bộ dữ liệu giống vậy cho các ngôn ngữ khác hay không phải cho văn bản phi cấu trúc. Các bộ dữ liệu khác nhau trình bày các tác vụ khác nhau cần giải quyết. Dưới đây là một vài bộ dữ liệu nổi tiếng:

- SVHN [18]: bộ dữ liệu về số nhà được trích xuất từ Google Street View. Các chữ số có nhiều hình dạng và kiểu viết khác nhau; tuy nhiên, mỗi số nhà được đặt ở giữa hình. Ảnh có độ phân giải không cao và cách sắp xếp của chúng có thể hơi kỳ lạ.
- ICDAR (2013 [19], 2015 [20]): 2 bộ dữ liệu dành cho hội nghị và cuộc thi ICDAR. Đây là 2 bộ dữ liệu benchmark tiêu chuẩn dùng để đánh giá rất nhiều mô hình phục vụ các tác vụ OCR cho ảnh có văn bản gần nằm ngang. Ngoài ra còn có các phiên bản cho những năm khác như 2003 hay 2019, tuy nhiên các bản năm 2013 và 2015 vẫn là phổ biến nhất.

- Total-Text [21]: tương tự các bộ dữ liệu ICDAR, đây cũng là một trong các bộ dữ liệu benchmark tiêu chuẩn, gồm các hình ảnh với nhiều loại văn bản khác nhau như các trường hợp văn bản ngang, nhiều hướng hay bị cong.

4.1.6. Một số công cụ mã nguồn mở

Trong một thời gian dài, Tesseract OCR [22] dẫn đầu các OCR tool mã nguồn mở, có thể nhận dạng hơn 100 ngôn ngữ. Tesseract 4 sử dụng các mô hình LSTM cho Text Recognition thay vì các phương pháp thị giác máy truyền thống như các phiên bản trước nhưng vẫn tỏ ra khá yếu với các loại dữ liệu phi cấu trúc và một nhược điểm khác là Tesseract chỉ được tối ưu cho CPU. Nhờ sự phát triển của Deep Learning, giờ đây ta đã có nhiều lựa chọn vượt trội hơn Tesseract, sử dụng được cho nhiều ngôn ngữ hay các trường hợp dữ liệu khác nhau, hỗ trợ cả việc huấn luyện lại (Retrain) hoặc tinh chỉnh (Fine-tuning). Chúng liên tục được phát triển và sử dụng các phương pháp tiếp cận hiện đại nhất cho cả 2 bài toán Detection và Recognition. Một số đại diện nổi bật có thể kể đến là docTR, keras-ocr, EasyOCR.

Ngoài ra, còn có các công cụ hỗ trợ việc sinh Synthetic text như SynthText [23] hỗ trợ việc “rắc” chữ một cách thông minh để chúng trông chân thật hay TRDG hỗ trợ tạo dữ liệu Synthetic text cho Text Recognition, đây cũng là trình tạo dữ liệu được dùng trong EasyOCR.

Cuối cùng, hội tụ của tất cả các công nghệ trên, PaddleOCR – một bộ công cụ hay nói đúng hơn là một hệ sinh thái cho OCR cực kỳ mạnh mẽ nhưng lại cực kỳ nhẹ (ultra lightweight), hỗ trợ đa ngôn ngữ cũng như nhiều thuật toán tiên tiến liên quan đến OCR và phát triển các mô hình hay giải pháp công nghiệp nổi bật: PP-Structure và PP-OCR [24] trên cơ sở này. Đồng thời, PaddleOCR cũng cung cấp các công cụ gán nhãn và tạo dữ liệu Synthetic, ngoài ra bộ công cụ này còn giúp huấn luyện và triển khai giữa các thiết bị máy chủ, di động hay các thiết bị nhúng và IoT. Đây cũng là lựa chọn mà chúng tôi sử dụng để triển khai một số phần của đề tài này. Ngoài ra, còn một lý do khác cho sự lựa chọn này là PaddleOCR được phát triển bởi các lập trình viên từ Trung Quốc nên sẽ có những sự tối ưu nhất định cho chữ Hán.

4.2. Các thành phần tính toán chính

Với sự phát triển hiện tại của học sâu (Deep Learning), các bài toán OCR nhận được nhiều giải pháp hơn. Sau đây là các thành phần cốt lõi giúp chúng tôi hình thành nên nhiều phương pháp tiếp cận cho bài toán OCR của mình, qua đó đạt được mục tiêu tăng tốc quá trình số hóa các văn bản Hán-Nôm.

4.2.1. Mạng Nơ-ron Tích chập (CNN)

Mạng Nơ-ron Tích chập (*Convolutional Neural Network - CNN*) là một họ các mạng nơ-ron mạnh mẽ được thiết kế với mục đích tận dụng các điểm ảnh kề cận thường có tương quan lẫn nhau để xây dựng những mô hình cho việc học từ dữ liệu ảnh hiệu quả hơn, thay vì chỉ đơn thuần loại bỏ cấu trúc không gian từ mỗi bức ảnh bằng cách chuyển chúng thành các vector và truyền qua một mạng kết nối đầy đủ.

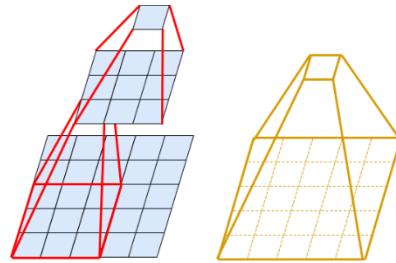
Các kiến trúc dựa trên CNN hiện nay xuất hiện khắp nơi trong lĩnh vực thị giác máy tính và được dùng trong nhiều bài toán như nhận dạng ảnh, phát hiện vật thể (Object Detection), phân vùng hình ảnh (Image Segmentation),... hoặc thậm chí cả các bài toán của lĩnh vực xử lý ngôn ngữ tự nhiên. CNN mượn rất nhiều ý tưởng từ ngành sinh học. Bên cạnh hiệu năng cao trên số lượng mẫu cần thiết để đạt được độ chính xác, CNN thường có hiệu quả tính toán tốt hơn và dễ thực thi song song trên nhiều GPU hơn các kiến trúc mạng fully connected [25].

4.2.1.1. Các khái niệm cơ bản

Các thành phần chính của CNN bao gồm: các lớp tích chập (Convolution), các chi tiết cơ bản quan trọng như việc sử dụng đa kênh (hay còn được gọi là các bộ lọc - filters) ở mỗi lớp, đệm (Padding) và sải bước (Stride) để điều chỉnh kích thước chiều của dữ liệu một cách hiệu quả, các lớp gộp (Pooling) dùng để giảm chiều và kết hợp thông tin qua các vùng không gian kề nhau. Cuối cùng là lan truyền qua lớp fully connected và một hàm kích hoạt để tính xác suất ảnh đó chứa vật thể gì.

Trong đó, lớp tích chập là quan trọng nhất và cũng là lớp đầu tiên của của mô hình CNN. Lớp này có chức năng phát hiện các đặc trưng có tính không gian một

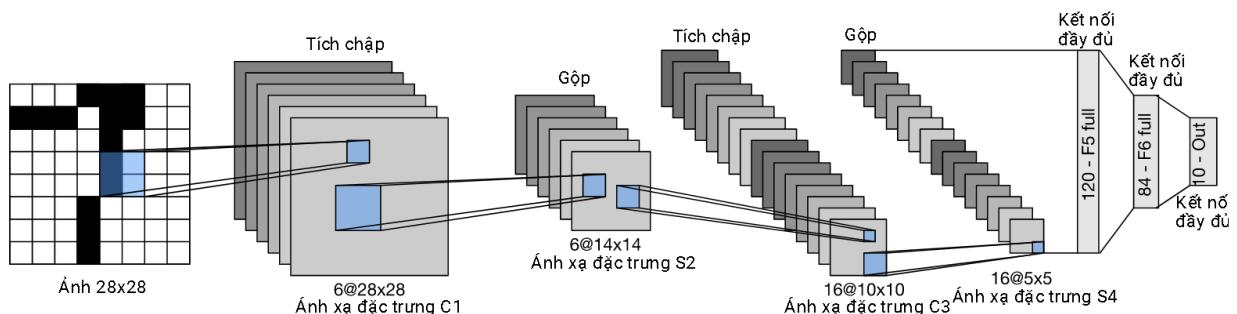
cách hiệu quả. Trong tầng này có 4 đối tượng chính là: ma trận đầu vào, các bộ lọc (filters), vùng nhận thức (receptive field), và feature map. Các filter sẽ trượt qua từng vị trí trên bức ảnh hay ma trận đầu vào để tính tích chập (convolution) giữa chúng và phần ảnh tương ứng. Phần tương ứng này trên bức ảnh gọi là receptive field, tức là vùng mà một neuron có thể nhìn thấy để đưa ra quyết định, và ma trận cho ra bởi quá trình này được gọi là feature map [26].



Hình 4.7. Vùng nhận thức (Receptive field)

Lấy ví dụ một ảnh đầu vào có kích thước 5×5 và một bộ lọc 3×3 . Từ hình trên có thể thấy, sau khi áp dụng bộ lọc trên ảnh đầu vào, giá trị của feature map sẽ được thể hiện trên một vùng 3×3 của ảnh. Nếu ta chuyển sang lớp thứ hai, bộ lọc 3×3 sẽ tiếp tục được áp dụng trên feature map này và ta thu sẽ được một giá trị duy nhất đại diện cho feature map hiện tại nói riêng và cho toàn bộ ảnh nói chung. Vì vậy, các feature map tiệm cận ảnh đầu vào sẽ có xu hướng tồn tại ít vùng nhận thức hơn và khi ta tiến tới các lớp cuối cùng, vùng nhận thức cũng sẽ to hơn.

Dưới đây là ảnh minh họa dòng dữ liệu trong LeNet 5 [27], một mạng CNN hoàn chỉnh đầu tiên, với đầu vào là một chữ số viết tay và đầu ra là xác suất với 10 kết quả khả thi:



Hình 4.8. Dòng dữ liệu trong LeNet (Nguồn [25])

4.2.1.2. Chuẩn hóa Batch (Batch Normalization)

Đối với các mạng tích chập, mô hình sẽ có găng học và điều chỉnh các trọng số để trích xuất các đặc trưng theo các mức độ khác nhau. Càng nhiều lớp hay mô hình càng sâu thì càng có thể trích xuất các đặc trưng ở mức cao hơn (high-level features). Huấn luyện một kiến trúc mạng sâu không hề đơn giản, hiệu suất của mô hình sẽ giảm xuống khi độ sâu mô hình tăng lên, đây được gọi là vấn đề suy thoái (degradation problem).

Chuẩn hóa theo batch (*Batch Normalization* - BN) [28] chính là một kỹ thuật phổ biến và hiệu quả nhằm tăng tốc độ hội tụ của mạng một cách ổn định. Cùng với các khối phần dư (residual block) sử dụng các kết nối tắt (skip connections), BN có thể giúp việc huấn luyện mạng học sâu với hơn 100 tầng một cách dễ dàng [29].

Trong quá trình huấn luyện (Training), chuẩn hóa theo batch sẽ áp dụng một phép biến đổi để duy trì trung bình của đầu ra gần với 0 và độ lệch chuẩn đầu ra gần 1, giúp các giá trị này ổn định hơn. Nghĩa là với mỗi kênh (channel) đang được chuẩn hóa, layer sẽ trả về kết quả:

$$BN(x) = \gamma \odot \frac{x - \hat{\mu}}{\hat{\sigma}} + \beta \quad (4.1)$$

Trong đó:

$\hat{\mu}$ là giá trị trung bình của các mẫu trong minibatch.

$\hat{\sigma}$ là độ lệch chuẩn của các mẫu trong minibatch.

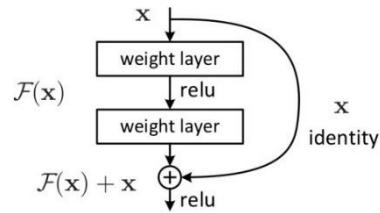
Vì việc lựa chọn đơn vị phương sai là tuỳ ý, nên hệ số tỷ lệ γ và độ chênh β thường được thêm vào và BN sẽ chủ động chuẩn hoá chúng theo giá trị trung bình μ và phương sai cho trước σ , nên độ lớn các giá trị kích hoạt ở những tầng trung gian không thể phân kỳ trong quá trình huấn luyện. Qua trực giác và thực nghiệm cho thấy, BN có thể cho phép chọn tốc độ học (learning rate) nhanh hơn.

BN hoạt động khác nhau trong quá trình huấn luyện và kiểm tra (Inference). Layer sẽ chỉ chuẩn hóa các đầu vào của nó trong quá trình kiểm tra sau khi đã được huấn luyện về dữ liệu có thống kê tương tự như dữ liệu kiểm tra [29].

4.2.1.3. Kết nối tắt (Skip connection)

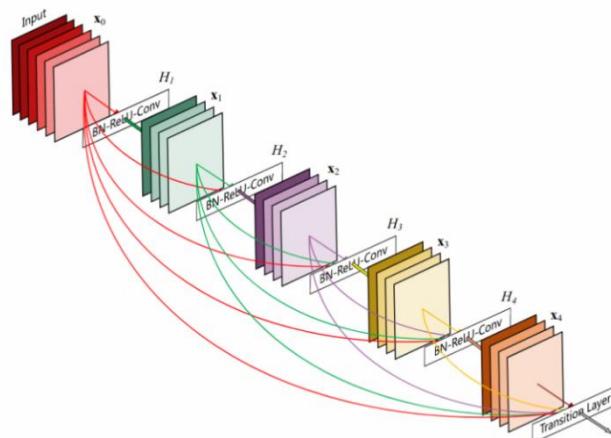
Kết nối tắt (Skip connection hay Shorcut connection) cũng là một cách khác giúp huấn luyện một mạng nơ-ron sâu, từ cái tên của nó ta cũng có thể thấy: bỏ qua một số lớp trong mạng và lấy đầu ra của một lớp làm đầu vào cho các lớp sau đó. Skip connections có thể được sử dụng theo 2 cách cơ bản trong mạng nơ-ron: phép cộng (Addition) và phép ghép (Concatenation) [30].

Trong trường hợp của Mạng phần dư (ResNet) [31], kết nối tắt giải quyết vấn đề suy thoái như đã đề cập ở trên. Thông tin từ các lớp ban đầu truyền đến các lớp sâu hơn bằng phép cộng ma trận, thao tác này không tốn thêm bất kỳ tham số nào.



Hình 4.9. Khối phần dư sử dụng kết nối tắt (Residual block, Nguồn [31])

Trong trường hợp của Mạng Tích chập Kết nối dày đặc (DenseNet) [32], kết nối tắt đảm bảo tính tái sử dụng các đặc trưng (feature reusability). Sự khác biệt chính giữa DenseNet và ResNet là DenseNet sẽ nối các feature map đầu ra của lớp này với lớp tiếp theo hơn là chỉ một phép tổng hợp. Nói cách khác DenseNet sử dụng phép ghép trong khi ResNet dùng phép cộng.



Hình 4.10. Khối dày đặc 5 lớp (5-layer dense block, Nguồn [32])

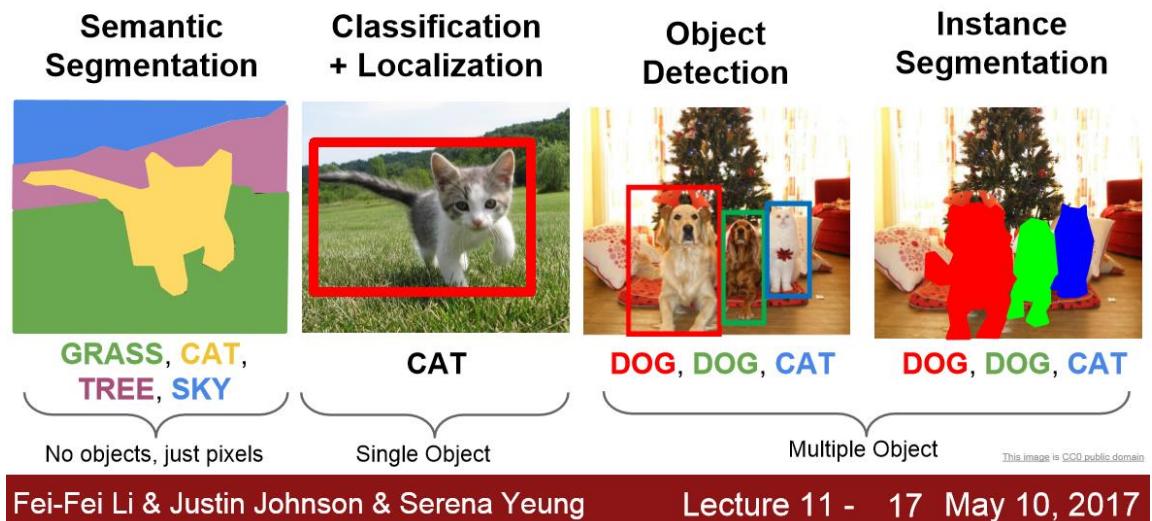
4.2.2. Phân vùng ảnh (Image Segmentation)

Trong các tác vụ phân loại ảnh đơn giản, ta chỉ quan tâm đến có được nhãn chung của tất cả các đối tượng có trong một hình ảnh. Trong phát hiện vật thể (Object Detection), ta đã tiến thêm một bước nữa bằng cách cố gắng biết tất cả các đối tượng hiện diện trong một ảnh cùng vị trí của chúng với sự trợ giúp của các khung chứa (bounding box - khung hình bao quanh vật thể).

Việc phân vùng hình ảnh (Image Segmentation) tiếp tục đưa ta tiến lên thêm một tầm cao mới bằng cách cố gắng tìm ra ranh giới chính xác (từng pixel) của các đối tượng trong ảnh [33].

4.2.2.1. Các khái niệm cơ bản

Other Computer Vision Tasks



Hình 4.11. Minh họa về Image Segmentation (Nguồn [34])

Một hình ảnh có thể xem là một tập hợp các pixel. Phân vùng ảnh là quá trình phân loại từng pixel đó về một lớp cụ thể và do vậy cũng có thể xem đây như là một bài toán phân loại trên mỗi pixel. Có 2 loại kỹ thuật phân vùng:

- Phân vùng theo ngữ nghĩa (Semantic segmentation): quá trình phân loại mỗi pixel về một nhãn cụ thể. Không phân biệt sự khác nhau giữa các object trong từng nhãn.

- Phân vùng theo cá thể (Instance segmentation): cung cấp một nhãn duy nhất cho mọi thể hiện của một object cụ thể trong ảnh. Như đã thấy trong hình trên, 3 object con vật đều được gán các nhãn (màu) khác nhau. Với Semantic segmentation, tất cả chúng sẽ được gán cùng màu.

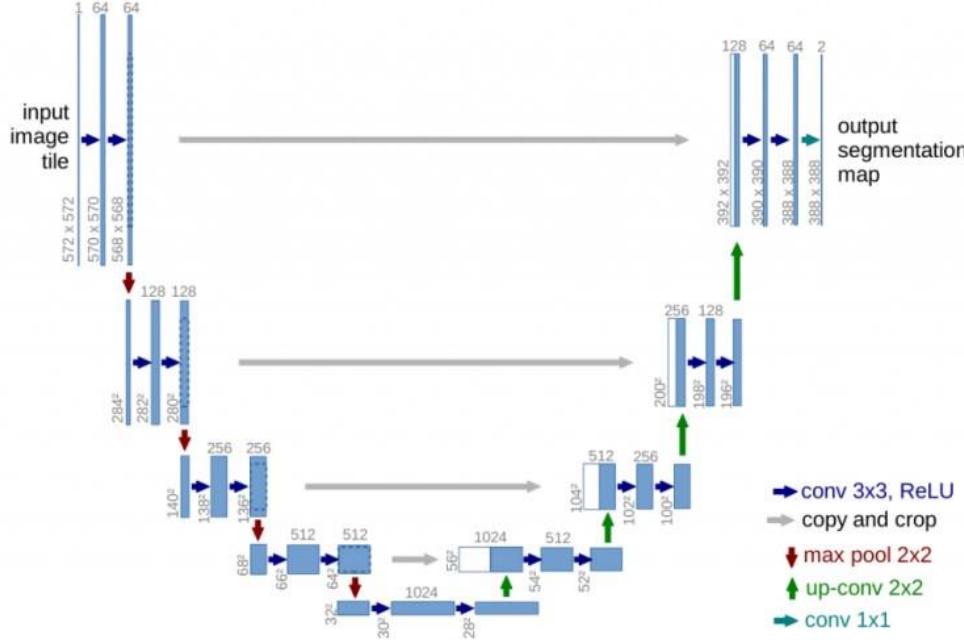
4.2.2.2. Ý tưởng từ mạng FCN

Kiến trúc chung của CNN thường gồm một vài lớp convolution và pooling, tiếp theo là vài lớp fully connected ở cuối. Bài báo về mạng tích chập hoàn toàn (*Fully Convolutional Network – FCN*) [35] đã lập luận rằng lớp fully connected cuối cùng này cũng có thể được coi là một phép tích chập 1×1 bao phủ toàn bộ khu vực. Nhờ vậy, các lớp fully connected cuối cùng có thể được thay thế bằng một lớp tích chập để đạt được kết quả tương tự. Lợi thế của việc làm này là kích thước đầu vào không cần cố định nữa. Khi dùng các lớp fully connected, kích thước đầu vào bị hạn chế; do đó, nếu một đầu vào có kích thước khác thì sẽ cần phải được resize lại. Nhưng bằng sự thay thế bởi lớp convolution, ràng buộc này sẽ không còn.

Vì feature map thu được ở lớp đầu ra đã bị thu hẹp (down-sample) do thực hiện phép tích chập nên ta sẽ cần gia tăng lại kích thước (up-sample) bằng kỹ thuật nội suy (interpolation). Nội suy song tuyến tính (bilinear interpolation) có thể giúp thực hiện điều này, nhưng bài báo trên đề xuất cách học up-sampling bằng việc sử dụng phép giải chập (deconvolution), điều này thậm chí có thể giúp học được cả việc gia tăng kích thước phi tuyến (non-linear up-sampling) [33].

4.2.2.3. Mạng U-Net

Mạng tích chập cho phân vùng hình ảnh y sinh (*Convolutional Networks for Biomedical Image Segmentation – U-Net*) [36] được xây dựng dựa trên mạng FCN. Nó bao gồm một bộ mã hóa sẽ down-sample hình ảnh đầu vào xuống thành một feature map (phản thu hẹp) và một bộ giải mã sẽ up-sample feature map đó lên kích thước hình ảnh đầu vào bằng cách sử dụng các lớp giải chập học được (phản mở rộng). Và sự đóng góp chính trong kiến trúc này chính là các kết nối tắt đối xứng giữa layer bên trái với layer bên phải.



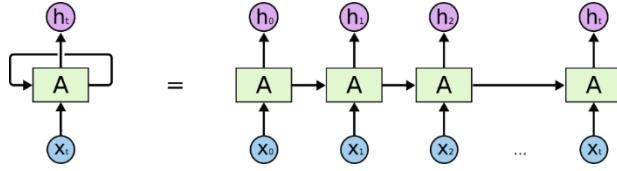
Hình 4.12. Kiến trúc U-Net (Nguồn [36])

4.2.3. Mạng Nơ-ron Hồi tiếp (RNN)

Mạng Nơ-ron Hồi tiếp (*Recurrent Neural Network – RNN*) là một mạng nơ-ron mạnh mẽ dùng để mô hình hóa dữ liệu chuỗi như chuỗi thời gian (time series) hay ngôn ngữ tự nhiên. RNN đặc biệt hữu ích cho việc đánh giá trình tự, theo đó các lớp ẩn (hidden layer) có thể học từ các lần chạy trước của mạng trên các phần trước đó trong trình tự hay nói cách khác các phần của mỗi lần chạy sẽ làm đầu vào lần chạy tiếp theo. Do đó, bên cạnh các mạng CNN có thể xử lý hiệu quả thông tin trên chiều không gian, thì các mạng RNN được thiết kế để xử lý thông tin tuần tự tốt hơn.

4.2.3.1. Các khái niệm cơ bản

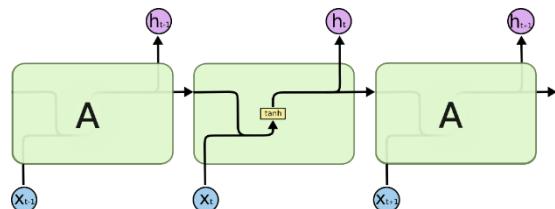
Các Mạng Nơ-ron Hồi tiếp là các mạng nơ-ron với các vòng lặp bên trong cùng các trạng thái ẩn (hidden state) để tổng hợp thông tin lịch sử của chuỗi cho tới bước thời gian (timestep) hiện tại. Cụ thể, kiến trúc này sử dụng đầu vào là một véc tơ x_t và đầu ra là một giá trị ẩn h_t . Chúng được đấu với một thân mạng nơ-ron A có tính chất truy hồi, cho phép thông tin được truyền từ bước này sang bước tiếp theo của mạng. Một mạng RNN có thể được coi là nhiều bản sao của cùng một mạng, mỗi bản sao truyền một thông điệp đến một mạng kế thừa [37].



Hình 4.13. Kiến trúc trải phẳng của RNN (Nguồn [37])

Một trong những điểm nổi bật của RNN là ý tưởng kết nối thông tin trước đó với nhiệm vụ hiện tại. Đôi khi, ta chỉ cần nhìn vào thông tin gần đây để thực hiện tác vụ hiện tại. Ví dụ như trong câu: "Những đám mây ở trên *bầu trời*", dường như trong một ngữ cảnh ngắn hạn, từ *bầu trời* có thể được dự đoán ngay mà không cần thêm các thông tin từ những câu văn khác gần đó. Tuy nhiên, có những tình huống ta sẽ cần nhiều ngữ cảnh hơn, chẳng hạn như: "Tôi lớn lên từ Việt Nam... Tôi nói *tiếng Việt*". Thông tin gần đó cho thấy rằng từ tiếp theo có thể là tên của một ngôn ngữ, nhưng nếu ta muốn thu hẹp phạm vi là ngôn ngữ nào, ta sẽ cần ngữ cảnh của các từ "Việt Nam" từ phía xa hơn trở lại, tức ta cần phải học để tìm ra từ *Việt* ở một ngữ cảnh dài hơn so với chỉ một câu. Những sự liên kết ngữ nghĩa dài này gọi là phụ thuộc dài hạn (long-term dependencies) [37].

Về lý thuyết, RNN có thể xử lý được những sự phụ thuộc trong dài hạn nhưng trên thực tế cho thấy RNN dường như học kém đi. Một trong những nguyên nhân chính được giải thích là do sự tiêu biến đạo hàm (vanishing gradient) khi trải qua chuỗi dài các tính toán truy hồi.



Hình 4.14. Module lặp lại trong một RNN chứa một lớp duy nhất (Nguồn [37])

4.2.3.2. Embedding từ (Word embedding)

Các mô hình học máy nhận đầu vào là các vector (các mảng số). Vì vậy, khi làm việc với dữ liệu văn bản, ta cần có chiến lược để chuyển các chuỗi thành số hay còn gọi là vector hóa (vectorize) văn bản trước khi đưa chúng vào mô hình. Ý tưởng

đầu tiên là ta có thể sử dụng biểu diễn one-hot (one-hot encoding). Để đại diện cho mỗi từ, ta sẽ tạo một vector có độ dài bằng số lượng từ vựng và tất cả các phần tử trong vector đó sẽ có giá trị bằng 0, sau đó gán giá trị 1 cho vị trí tương ứng với từ đó trong vector one-hot này.

Nhưng cách tiếp cận trên có thể không hiệu quả. Một vector one-hot là có thể rất thưa thớt, tức là hầu hết các chỉ số đều bằng 0. Nếu ta có 10.000 từ trong kho từ vựng, để mã hóa one-hot cho từng từ, ta sẽ cần tạo một vector trong đó có 99,99% các phần tử bằng 0.

	cat	mat	on	sat	the
the =>	0	0	0	0	1
cat =>	1	0	0	0	0
sat =>	0	0	0	1	0
...					...

Hình 4.15. One-hot encoding (Nguồn [38])

Embedding từ (*Word embedding*) giúp khắc phục hạn chế trên, cho phép ta sử dụng một biểu diễn hiệu quả, dày đặc (dense), trong đó các từ giống nhau được mã hóa tương tự nhau. Quan trọng hơn hết là ta không cần chỉ định các giá trị mã hóa này một cách thủ công mà chúng sẽ là các trọng số có thể được học bởi mô hình.

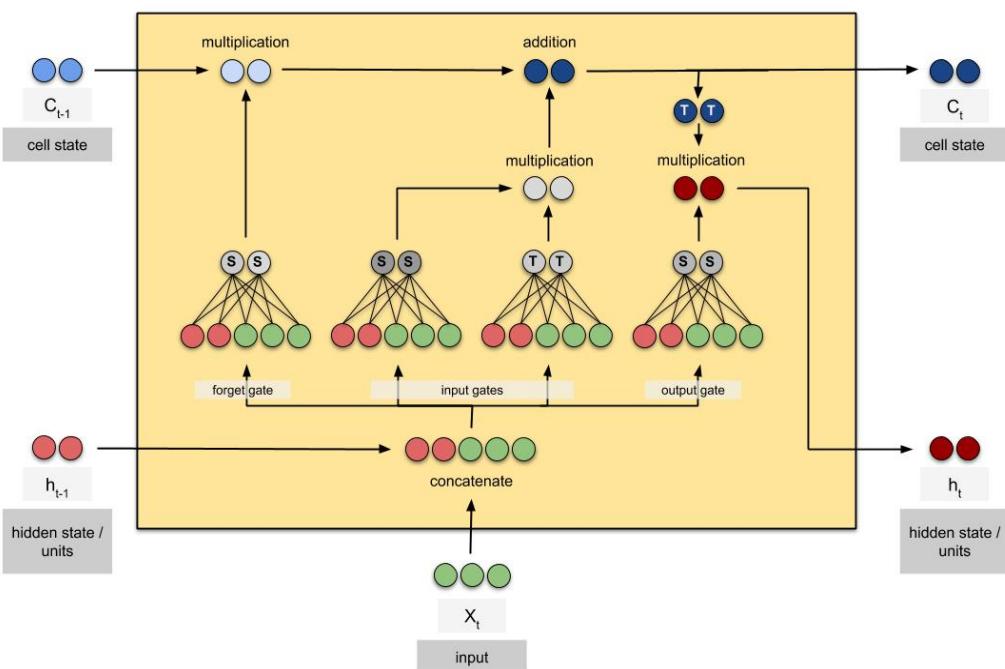
Một embedding nhiều chiều hơn có thể nắm bắt một cách chi tiết hơn các mối quan hệ giữa các từ nhưng cũng sẽ cần nhiều dữ liệu hơn học. Cũng có thể xem embedding như là một "bảng tra cứu". Sau khi đã học xong các trọng số, ta có thể mã hóa từng từ bằng cách tra cứu vector dày đặc trong bảng tương ứng với nó [38].

cat =>	1.2	-0.1	4.3	3.2
mat =>	0.4	2.5	-0.9	0.5
on =>	2.1	0.3	0.1	0.4
...				

Hình 4.16. Một embedding 4 chiều (Nguồn [38])

4.2.3.3. Nút Hồi tiếp có Cổng (GRU)

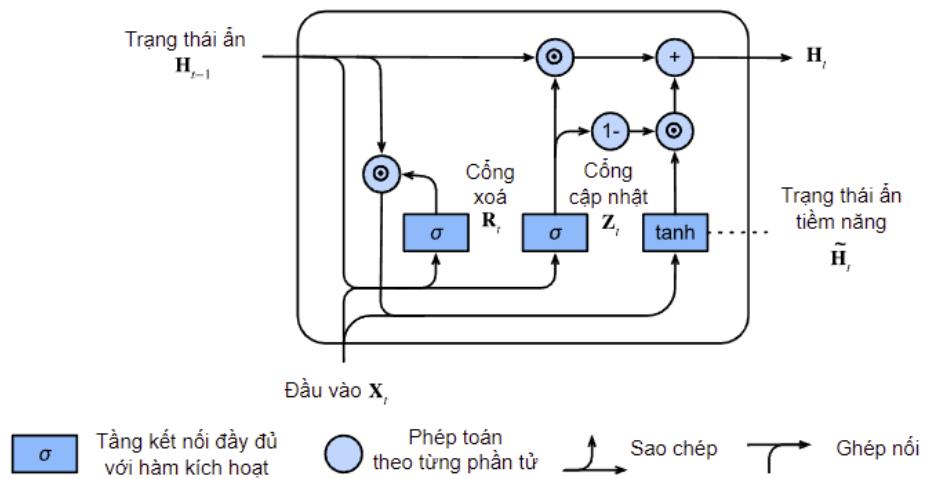
Một trong những phương pháp ra đời sớm nhất để giải quyết vấn đề long-term dependencies đã đề cập ở trên là Bộ nhớ ngắn hạn dài (*Long Short-Term Memory - LSTM*) [39] được lấy cảm hứng từ các cổng logic của máy tính. LSTM đưa ra một khái niệm là ô nhớ (memory cell) để ghi lại thông tin bổ sung. Để kiểm soát một ô nhớ, ta cần một số cổng: cổng đầu ra (output gate) để đọc các thông tin từ ô nhớ đó, cổng đầu vào (input gate) để quyết định thời điểm đọc dữ liệu vào ô nhớ, cổng quên (forget gate) để thiết lập lại nội dung ô nhớ [40].



Hình 4.17. Một khối LSTM với 2 nút ẩn và 1 đầu vào 3 chiều (Nguồn [41])

Nút Hồi tiếp có Cổng (*Gated Recurrent Unit - GRU*) [42] là một biến thể đơn giản hơn của LSTM nhưng thường có chất lượng tương đương và tính toán nhanh hơn đáng kể. GRU hợp nhất ô trạng thái (cell state) và trạng thái ẩn (hidden state), đồng thời sử dụng một cổng xóa (reset gate) R_t chịu trách nhiệm cho việc nắm bắt phụ thuộc ngắn hạn. Khi cổng xóa được kích hoạt (có giá trị gần với 1), GRU trở thành RNN thông thường. Nếu tất cả các phần tử của cổng xóa gần 0, trạng thái ẩn tiềm năng (candidate hidden state) sẽ là đầu ra của một perceptron đa tầng (MLP) với đầu vào là X_t .

Do đó, mọi trạng thái ẩn trước đó sẽ được đặt về lại giá trị mặc định. GRU cũng kết hợp cổng quên và cổng đầu vào trong LSTM thành một cổng cập nhật (update gate) Z_t chịu trách nhiệm cho việc ghi nhớ dài hạn. Nếu các giá trị trong cổng này bằng 1, ta đơn giản giữ lại trạng thái cũ. Ngược lại, nếu Z_t gần với 0, trạng thái ẩn sẽ gần với trạng thái ẩn tiềm năng.



Hình 4.18. Tính toán trạng thái ẩn trong GRU (Nguồn [43])

Mô hình GRU có được cuối cùng đơn giản hơn các mô hình LSTM tiêu chuẩn và trở nên ngày càng phổ biến [43]. Tuy vậy, với một chuỗi đầu vào rất dài, vấn đề long-term dependencies do sự tiêu biến hoặc bùng nổ đạo hàm (vanishing/exploding gradient) vẫn chưa thực sự được giải quyết.

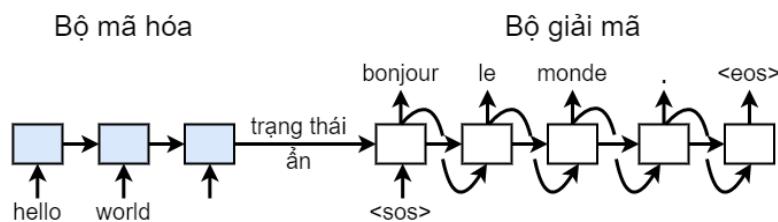
4.2.3.4. Mạng Nơ-ron Hồi tiếp 2 chiều

Đối với các chuỗi trình tự không phải là time series (ví dụ: văn bản), RNN thường có thể hoạt động tốt hơn nếu nó không chỉ xử lý trình tự từ đầu đến cuối mà còn xử lý ngược lại. Ví dụ, để dự đoán từ tiếp theo trong một câu, việc có ngữ cảnh xung quanh từ đó thường sẽ hữu ích hơn là chỉ giới hạn trong những từ đứng trước nó. Mạng Nơ-ron Hồi tiếp 2 chiều (*Bidirectional RNN*) sẽ lan truyền đầu vào về phía trước và ngược lại thông qua lớp RNN, sau đó đầu ra cuối cùng sẽ được nối lại với nhau. Nhờ vậy, ta sẽ có được các quan sát cả trong tương lai và quá khứ để dự đoán hiện tại. Trạng thái ẩn tại mỗi bước thời gian được xác định đồng thời bởi thông tin ở trước và sau bước thời gian đó.

Việc học RNN 2 chiều rất tốn kém do các chuỗi đạo hàm dài, ngoài ra tuy nói rằng “hai chiều” nhưng bản chất đây chỉ là 2 mạng RNN chạy độc lập, không liên quan gì nhau nên vẫn chưa thực sự biểu diễn được cái gọi là “thông tin 2 chiều” của một từ bằng một cách có liên quan đến cả từ trước lẫn từ sau, tức liên quan đến tất cả các từ còn lại trong câu.

4.2.3.5. Mô hình chuỗi sang chuỗi (Seq2Seq)

Mô hình chuỗi sang chuỗi (*Sequence to Sequence – Seq2Seq*) đã đạt được nhiều thành công trong các tác vụ như dịch máy, tóm tắt văn bản hay sinh mô tả cho ảnh. Các kiến trúc này lần đầu được giải thích trong 2 bài báo tiên phong [44] và [45], sử dụng kiến trúc mã hóa - giải mã (Encoder – Decoder) để sinh ra chuỗi đầu ra từ chuỗi đầu vào (có thể là từ, ký tự hoặc các đặc trưng của ảnh).



Hình 4.19. Kiến trúc mô hình chuỗi sang chuỗi (Seq2Seq)

Bộ mã hóa sẽ xử lý thông tin của chuỗi đầu vào với độ dài khác nhau và biến đổi thông tin thu được thành một vector ngữ cảnh (context vector). Sau khi xử lý toàn bộ chuỗi đầu vào, bộ mã hóa sẽ gửi vector ngữ cảnh này tới bộ giải mã để từ đó bắt đầu hình thành thông tin cho chuỗi đầu ra. Bộ mã hóa và bộ giải mã thường có xu hướng là các mạng RNN, tại mỗi bước thời gian, các RNN này sẽ thực hiện một số xử lý và cập nhật trạng thái ẩn dựa trên các đầu vào của chúng và các đầu vào trước đó mà chúng đã thấy [46].

Một trong những nhược điểm của phương pháp này là nút thắt cổ chai phát sinh với việc sử dụng vector mã hóa có độ dài cố định, theo đó thông tin được cung cấp bởi đầu vào mà Decoder nhận được sẽ bị hạn chế. Với các trình tự dài, phức tạp, kích thước biểu diễn của chúng sẽ bị buộc phải giống như với các trình tự ngắn hoặc đơn giản hơn và sự ra đời của Cơ chế Tập trung đã giúp giải quyết vấn đề này.

4.2.4. Cơ chế Tập trung (Attention Mechanism)

Cơ chế Tập trung (*Attention Mechanism*) [47], một khái niệm cách mạng đang thay đổi cách mà ta áp dụng Deep Learning, được giới thiệu như một giải pháp để cải thiện hiệu suất của mô hình Encoder – Decoder cho dịch máy. Đây là một trong những đột phá có giá trị nhất trong nghiên cứu học sâu trong thập kỷ qua, tạo tiền đề cho rất nhiều đột phá gần đây trong lĩnh vực xử lý ngôn ngữ tự nhiên.

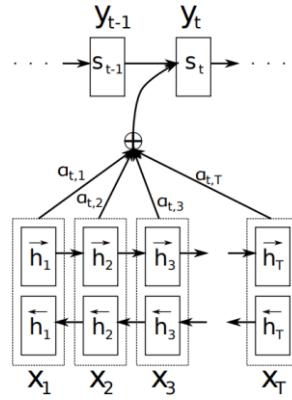
Cơ chế này cho phép Decoder tận dụng các phần có liên quan nhất trong chuỗi đầu vào một cách linh hoạt bằng một sự kết hợp có trọng số của tất cả các vector đầu vào được mã hóa với các vector phù hợp nhất có trọng số cao nhất. Hay rộng hơn, cơ chế này nhằm mục đích loại bỏ sự bất lợi do việc nén và mất thông tin trong RNN. Điều này bắt nguồn do việc mã hóa với độ dài cố định của các trạng thái ẩn, có nguồn gốc từ các chuỗi đầu vào tạo bởi các lớp hồi tiếp trong mô hình Seq2Seq.

4.2.4.1. Khởi nguồn

Trước đó, các mô hình dịch máy dựa trên các kiến trúc Encoder – Decoder sử dụng các mạng RNN. Hạn chế chính của hướng tiếp cận này là nếu Encoder cho ra một kết quả không tốt khi nó cố gắng hiểu các câu dài hơn, thì bản dịch lúc này cũng sẽ cho ra kết quả tệ. Đây là vấn đề phụ thuộc dài hạn đã đề cập của các mạng RNN nói chung.

Ngay cả tác giả của [45], người đề xuất mạng Encoder – Decoder, cũng đã chứng minh rằng hiệu suất của cách làm này suy giảm nhanh chóng khi độ dài của câu đầu vào tăng lên. Một vấn đề khác nữa là không có cách nào để đánh giá tầm quan trọng của một số từ đầu vào so với những từ khác trong khi dịch câu.

Bahdanau cùng các cộng sự đã đề xuất một ý tưởng đơn giản nhưng hiệu quả [47], theo đó tác giả đề xuất rằng không những có thể quan sát tất cả các từ đầu vào trong vector ngữ cảnh, mà các thông tin quan hệ quan trọng của từng từ cũng sẽ được xem xét. Vì vậy, bất cứ khi nào mô hình được đề xuất tạo ra một câu, nó sẽ tìm kiếm một tập hợp các vị trí trong các trạng thái ẩn của Encoder, nơi có thông tin phù hợp nhất. Ý tưởng này được gọi là Attention [48].

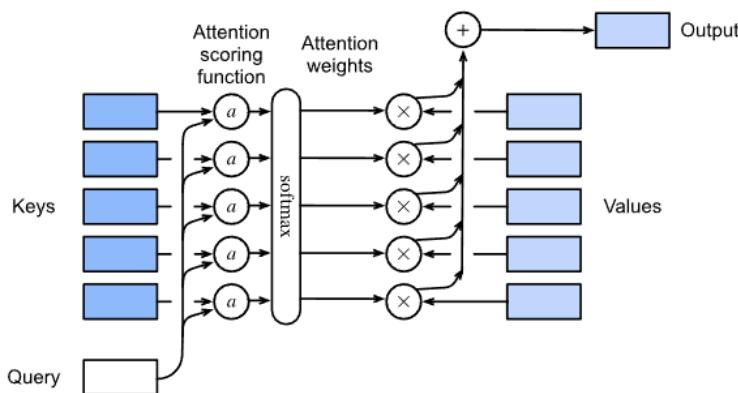


Hình 4.20. Mô hình sinh từ thứ t từ câu đầu vào sử dụng Attention (Nguồn [47])

4.2.4.2. Các tính toán chính

Cơ chế Tập trung có thể được xem là phép gộp tổng quát theo trọng số trên mỗi giá trị đầu vào gồm 3 thành phần chính: truy vấn Q (query) tương tự như giá trị đầu ra trước đó của Decoder, khóa K (key) và giá trị V (value) tương tự với các giá trị đầu vào được mã hóa.

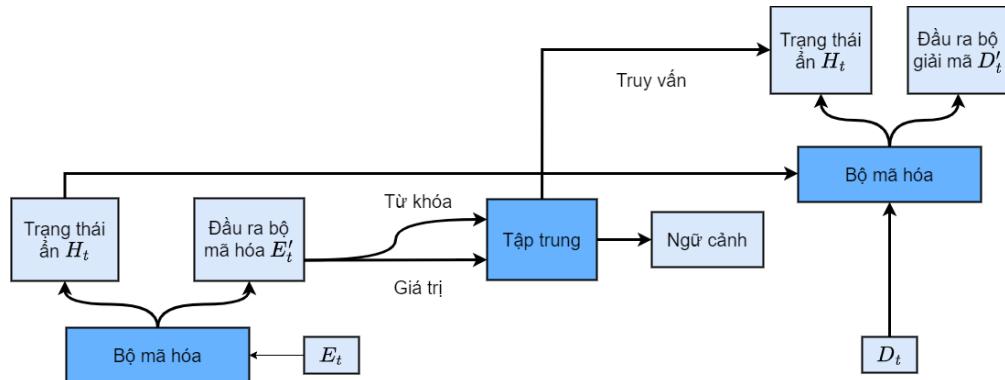
Với mỗi câu truy vấn q , tầng tập trung sẽ trả về đầu ra dựa trên bộ nhớ là các cặp khóa-giá trị $(k_1, v_1), \dots, (k_n, v_n)$ được mã hóa trong tầng tập trung này. Và với mỗi cặp khóa-giá trị đó, ta sử dụng một hàm tính điểm α được tính bằng công thức: $a_i = \alpha(q, k_i)$, để đo độ tương đồng giữa câu truy vấn và các khóa, từ đó tính toán đầu ra của tầng tập trung. Tiếp theo, một hàm softmax được sử dụng để thu được các trọng số tập trung (attention weights): $b = \text{softmax}(a)$. Cuối cùng, đầu ra của tầng tập trung là tổng trọng số của các giá trị $o = \sum_{i=1}^n b_i v_i$ [49].



Hình 4.21. Tính toán đầu ra của tầng tập trung (Nguồn [49])

4.2.4.3. Seq2Seq sử dụng Cơ chế Tập trung

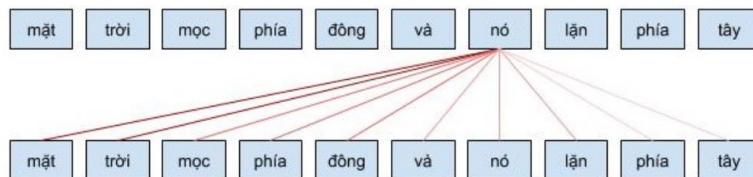
Bộ nhớ tầng tập trung ở đây bao gồm tất cả thông tin mà Encoder học được, tức đầu ra của Encoder tại từng bước thời gian t có cùng các cặp khóa và giá trị. Trạng thái ẩn của Encoder tại bước thời gian cuối cùng sẽ là trạng thái ẩn ban đầu của Decoder. Trong khi đó, đầu ra của Decoder tại bước thời gian trước đó $t-1$ được sử dụng làm câu truy vấn. Đầu ra của mô hình có thể được hiểu là thông tin ngữ cảnh của chuỗi được ghép nối với đầu vào của Decoder D_t .



Hình 4.22. Mô hình Seq2Seq sử dụng cơ chế Attention

4.2.4.4. Tự tập trung (Self-Attention)

Cơ chế Tự tập trung (*Self-Attention*) cho phép mô hình khi mã hóa một từ có thể sử dụng thông tin của những từ liên quan tới nó. Tương tự như các mô hình Attention thông thường, mô hình Self-Attention cũng có câu truy vấn, khóa và giá trị nhưng chúng được sao chép từ các phần tử trong chuỗi đầu vào, tầng Self-Attention sẽ trả về một chuỗi có cùng độ dài với chuỗi đầu vào đó.

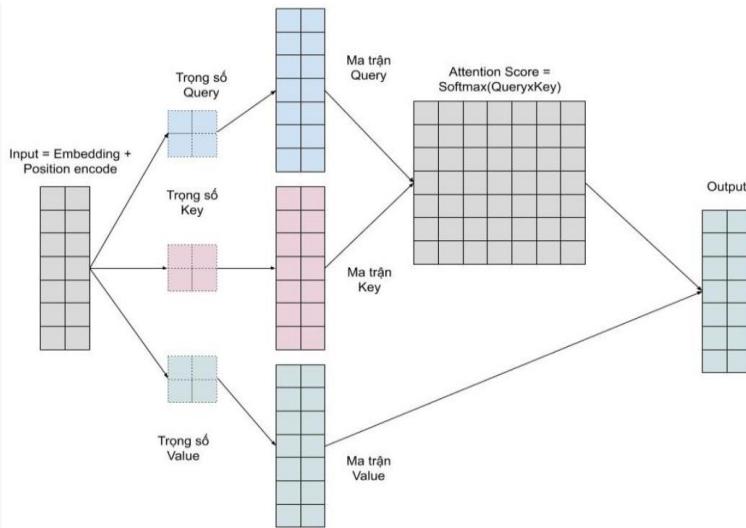


Hình 4.23. Một từ sẽ chú ý tới các từ liên quan trong Self-Attention (Nguồn [50])

Vì các truy vấn, khóa và giá trị đều đến từ cùng một nơi, điều này biểu diễn sự “Tự tập trung”. Self-Attention có thể được tính toán song song so với RNN và cũng có thể xem nó như là một cơ chế tìm kiếm với:

- Vector truy vấn (query): chứa thông tin từ được tìm kiếm. Giống như câu truy vấn của Google search.
- Vector khóa (key): biểu diễn thông tin các từ được so sánh với từ cần tìm kiếm ở trên. Giống như các trang web được so sánh với từ khóa được tìm.
- Vector giá trị (value): thể hiện nội dung, ý nghĩa của các từ. Giống như nội dung trang web được hiển thị cho người dùng sau khi tìm kiếm.

Ta dùng tích vô hướng để tính chỉ số tương quan giữa ma trận của vector truy vấn và ma trận của vector giá trị, sau đó dùng một hàm softmax để đưa chỉ số vừa tính được về dạng xác suất. Kết quả cuối cùng sẽ là trung bình có trọng số (weighted average) giữa vector giá trị và chỉ số tương quan. Phép tính này còn gọi là *Dot-Product Attention*. Như vậy, khác với cơ chế Attention thông thường cho phép đầu ra tập trung sự chú ý vào đầu vào trong quá trình sinh đầu ra, còn Self-Attention sẽ cho phép các đầu vào tương tác với nhau [50].



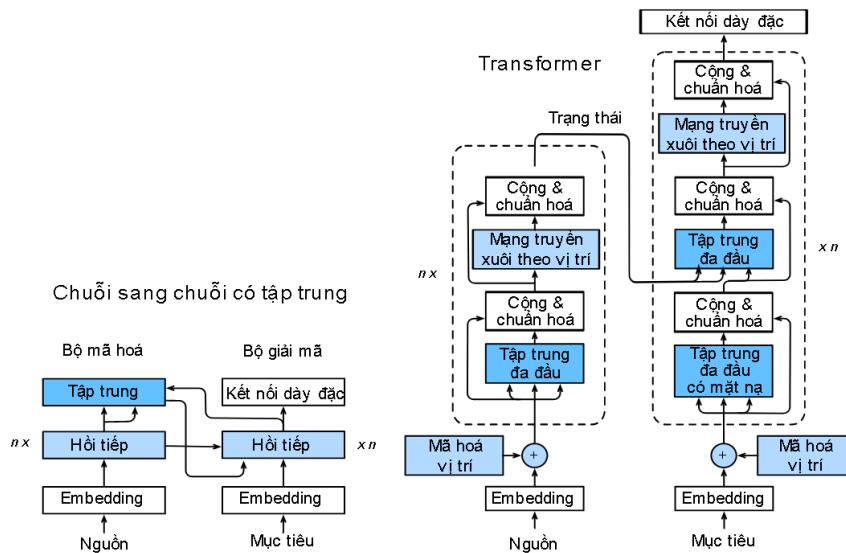
Hình 4.24. Dot-Product Attention (Nguồn [50])

4.2.4.5. Kiến trúc Transformer

Transformer [51] đã trở nên phổ biến rộng rãi trong một loạt các ứng dụng Deep Learning hiện đại như trong các lĩnh vực ngôn ngữ, thị giác, giọng nói hay học tăng cường. Nói một cách đơn giản, Transformer chỉ dựa trên cơ chế Attention, nhưng cũng ở mức tinh vi hơn khi so với cách được đề xuất trước đó [52].

Ý tưởng cốt lõi đằng sau một mô hình Transformer là Self-Attention, khả năng tập trung đến các vị trí khác nhau của trình tự đầu vào để tính toán biểu diễn cho trình tự đó. Transformer có thể xử lý đầu vào có kích thước thay đổi bằng cách sử dụng ngăn xếp (stack) các lớp Self-Attention thay vì các mạng RNN hay CNN. Kiến trúc chung này có một số ưu điểm [53]:

- Nó không có khái niệm về mối quan hệ thời gian hay không gian trên dữ liệu, nên tạo điều kiện lý tưởng để xử lý một tập hợp các đối tượng nhưng đây đồng thời cũng là nhược điểm. Nếu đầu vào tồn tại các mối quan hệ này như là dữ liệu văn bản, ... Lúc này, một số lớp biểu diễn vị trí (Positional Encoding) cần được thêm vào.
- Đầu ra của các lớp có thể tính toán song song thay vì tuần tự như RNN.
- Các thông tin ở xa có thể ảnh hưởng đến đầu ra của nhau mà không cần phải băng qua nhiều bước RNN hoặc các lớp tích chập.
- Nó có thể học được các sự phụ thuộc dài hạn, đây là một thử thách trong rất nhiều tác vụ trình tự.



Hình 4.25. Kiến trúc Transformer so với Attention-Seq2Seq (Nguồn [54])

Có thể thấy trong hình trên, Transformer cũng bao gồm một Encoder và một Decoder. Khác với mô hình Seq2Seq, embedding biểu diễn chuỗi đầu vào và đầu ra sẽ được cộng thêm với thông tin vị trí thông qua Positional Encoding trước khi

được đưa vào Encoder và Decoder để xếp các module dựa trên Self-Attention. một Tầng hồi tiếp trong Seq2Seq cũng được thay bằng các Khối Transformer tương ứng. Trong đó, Transformer Encoder là một ngăn xếp gồm nhiều lớp Encoder giống nhau, trong đó mỗi lớp sẽ có các lớp con, bao gồm:

- Một tầng Tập trung Đa đầu (Multi-Head Attention): đây là một layer mới được giới thiệu trong bài báo [51], tạo nên sự khác biệt với các mô hình RNN bằng việc sử dụng nhiều Self-Attention, giúp mô hình có thể học nhiều kiểu mối quan hệ giữa các từ với nhau.
- Một Mạng truyền Xuôi theo Vị trí (Position-wise Feed-Forward Network): gồm 2 lớp fully connected với một hàm kích hoạt ReLU ở giữa.
- Các Kết nối tắt (Skip Connection) và lớp Chuẩn hóa theo Tầng (Layer Normalization): giúp mô hình nhanh hội tụ hơn và trách mất mát thông tin trong quá trình huấn luyện.

Transformer Decoder cũng là một ngăn xếp gồm nhiều lớp giống nhau và khá giống kiến trúc của Encoder, thực hiện chức năng giải mã vector của câu nguồn thành câu đích. Khối này có thêm một tầng Multi-Head Attention khác nằm ở giữa để nhận vào trạng thái của Encoder và học mối liên quan giữ từ đang dịch với các từ ở câu nguồn. Cụ thể hơn, các câu truy vấn của riêng tầng này sẽ là các đầu ra của lớp Decoder trước đó, còn các cặp khóa và giá trị là từ các đầu ra của Transformer Encoder. Còn trong Self-Attention của Decoder, các truy vấn, cặp khóa và giá trị đều từ đầu ra của lớp Decoder trước đó.

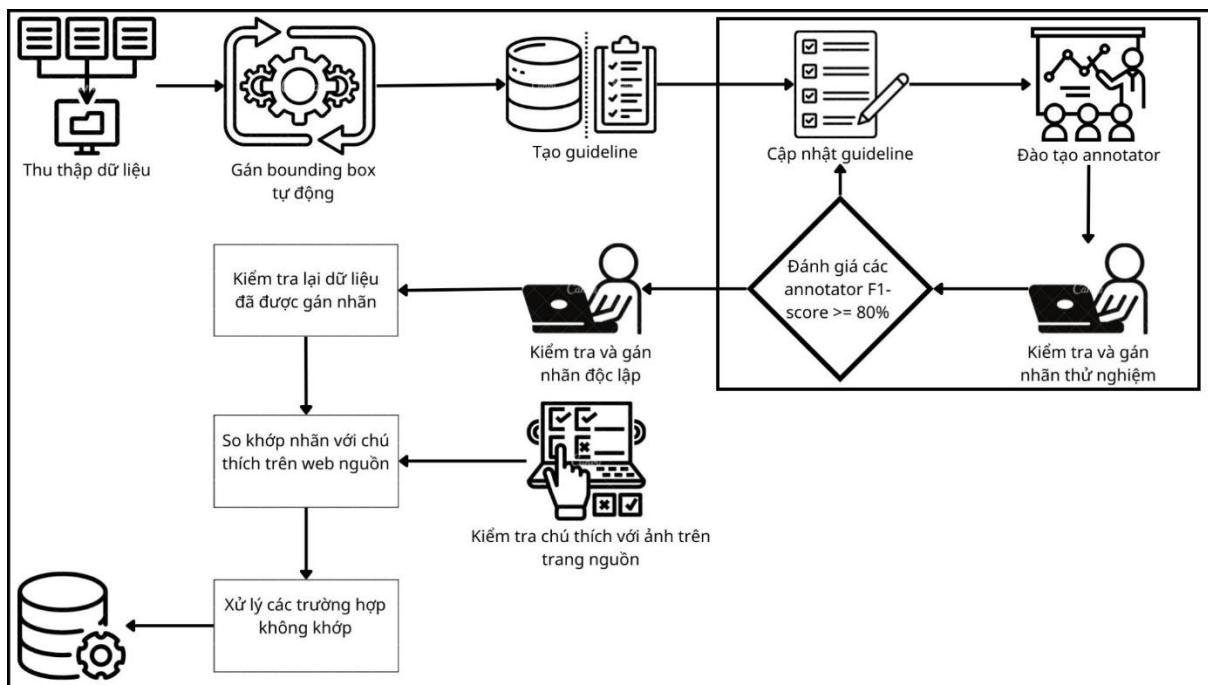
Tuy nhiên, với mỗi vị trí trong Decoder ta cần che đi (mask) các từ ở tương lai chưa được mô hình dịch đến, nghĩa là chỉ có thông tin từ vị trí đầu tiên và thứ hai sẽ được sử dụng để dự đoán cho vị trí thứ ba. Tương tự, để dự đoán thông tin cho vị trí thứ tư, chỉ thông tin từ vị trí đầu tiên, thứ hai và thứ ba sẽ được sử dụng, ... Attention được che này (Masked Attention) bảo toàn khả năng tự động hồi quy (auto regressive), đảm bảo rằng dự đoán chỉ phụ thuộc vào các thông tin đầu ra đã được tạo [54].

Chương 5. BỘ DỮ LIỆU NomNaOCR

5.1. Khái quát chung

Vì đây là bộ dữ liệu đặc thù dành cho chữ Hán-Nôm cũ nên trước tiên chúng tôi cần phải tiến hành tìm kiếm nguồn dữ liệu uy tín nhằm mục đích tăng độ tin cậy cho bộ dữ liệu này sau khi hoàn thành. Chúng tôi lựa chọn trang web của Hội Bảo tồn di sản chữ Nôm Việt Nam – VNPF [1] làm nguồn dữ liệu duy nhất vì đây là một hội lớn về chữ Nôm ở Việt Nam với sự cố vấn của nhiều giáo sư, tiến sĩ trong và ngoài nước với mục đích là bảo tồn các di sản văn hóa được ghi lại dưới dạng chữ Nôm đang dần bị mai một của dân tộc bằng việc phát triển những công cụ phần mềm để làm việc số hóa, in ấn, nghiên cứu, bảo quản vật lí, và chia sẻ trên Internet nhiều tác phẩm trong bộ sưu tập thư viện quốc gia và đền chùa Phật giáo.

Sau gần 20 năm, VNPF đã làm việc để thúc đẩy khảo cứu các văn bản Hán-Nôm và nền văn hóa được tạo ra trong công chúng. Quy trình xây dựng bộ dữ liệu của chúng tôi bằng phương pháp bán thủ công được mô tả như hình dưới, và sẽ đề cập chi tiết trong các mục bên dưới.



Hình 5.1. Quy trình xây dựng bộ dữ liệu NomNaOCR

5.2. Thu thập dữ liệu

VNPF [1] đã số hóa cho rất nhiều tác phẩm Hán-Nôm nổi tiếng có giá trị lịch sử cao, từ đó cung cấp cho cộng đồng bản quét ảnh của nhiều trang sách viết tay cùng với các ký tự Hán-Nôm kỹ thuật số tương ứng như hình dưới. Để có thể sử dụng được khôi tài nguyên vô giá trên, chúng tôi sử dụng một tiện ích có tên là Automa [55] để tạo một luồng thu thập tự động cho nguồn dữ liệu này: từ các hình ảnh län URL của chúng cho tới các nội dung được phiên dịch gồm các ký tự Hán-Nôm kỹ thuật số và phần dịch Quốc ngữ của chúng (nếu có).

Tục biên tự [12 trang]

Tách câu và Phiên âm

大越史記續編序 . [1a*1*1]

Đại Việt sử kí tục biên tự.

國之有史尚矣 . [1a*2*1]

Quốc chi hữu sử thượng hĩ.

我越歷代史記先正黎文休潘孚先作
之於前吳士連武瓊述之於後 . [1a*2*7]

Ngã Việt, lịch đại sử kí tiên chính Lê Văn Hưu, Phan Phu Tiên tác chí ư tiên, Ngô Sĩ Liên, Vũ Quỳnh thuật chí ư hậu.

其間事蹟之詳畧政治之得失莫不悉
備於記載之中 . [1a*4*8]

Ki gian sự tích chí tướng lược, chính trị chí đắc thất,
mặc bất tất bị ư kí tài chí trung.

但未行鋟梓更手傳筆因循抄錄不能
無陶陰帝廟之疑 . [1a*6*3]

Đãn vị hành tẩm tử, cành thủ truyền bút, nhân tuân
sao lục, bất năng vô đào âm để hổ chí nghi.

迨至 . [1a*7*12]

Đãi chí ...

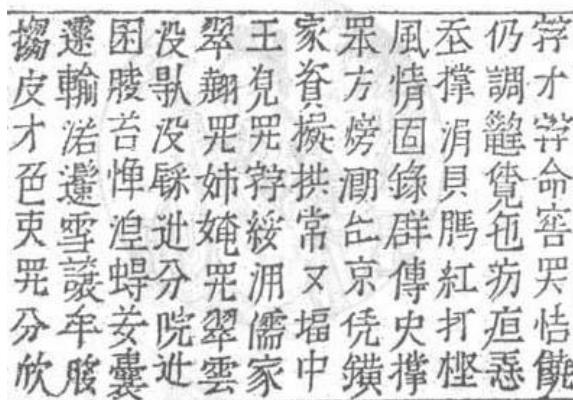
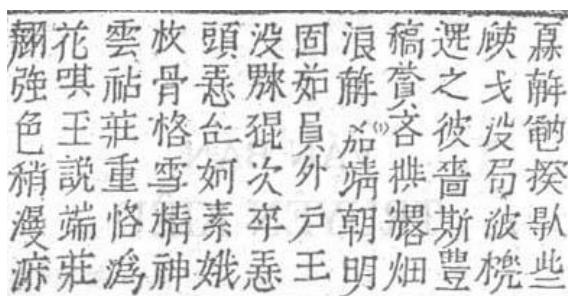


Trang: 1a

Hình 5.2. Toàn thư, Quyển thứ, tờ 1a trên website của VNPF

Chúng tôi lựa chọn các tập thơ nổi tiếng như Lục Vân Tiên của cụ Đò Nguyễn Đình Chiểu, Truyện Kiều (bản chép tay thuộc các năm 1866, 1871 và 1872) của Đại thi hào Nguyễn Du. Ngoài ra, còn có Đại Việt Sử Ký Toàn Thư (ĐVSKTT), một bộ Quốc sử lớn của dân tộc dưới dạng văn xuôi, do nhiều nhà sử học từ đời nhà Trần

cho tới nhà Hậu Lê biên soạn. Mục đích của việc thu thập thêm lượng dữ liệu dòi dào của tác phẩm văn xuôi này là làm gia tăng sự phong phú, đa dạng cho bộ dữ liệu, vì các tập thơ được sử dụng là thể lục bát với các câu 6 và 8 ký tự đan xen nhau, do đó cấu trúc các ký tự xuất hiện trong ảnh rõ ràng và dễ dàng xác định hơn ([hình 5.3](#)). Trong khi đó văn xuôi có số lượng các ký tự trong mỗi câu khác nhau và không tuân theo một quy luật cụ thể nào, dẫn đến cấu trúc các ký tự trong ảnh cũng khác nhau ([hình 5.4](#)).



Hình 5.3. Cấu trúc thường thấy
trong các tác phẩm thơ lục bát

○ 卷之四	屬吳晉宋齊梁紀	起癸卯至丙寅九	屬東漢紀	起癸卯至丙寅九	屬西漢紀	起辛未至己亥九
前季紀	起辛酉至丁卯	在位四十年	士王紀	起丁卯至丙寅	徵女王紀	起庚子至壬寅
九年	至丁	在位三十年	士王	至丙	三年	在位三年
七年						

Hình 5.4. Cấu trúc bất định trong
Đại Việt Sử Kí Toàn Thư

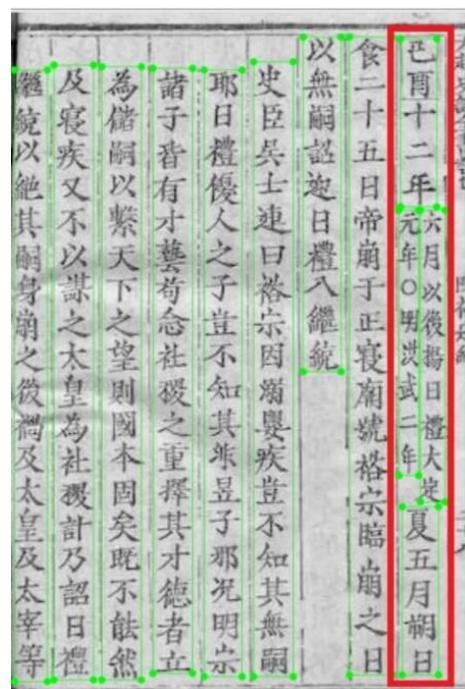
Kết thúc giai đoạn thu thập dữ liệu, chúng tôi có được 2956 trang sách viết tay (Page) cho toàn bộ dữ liệu và số lượng Page cụ thể cho từng tác phẩm theo bảng 5.1, trong đó Truyện Kiều bản 1866 có số lượng Page ít nhất với 100 Page và Bản kỷ của Đại Việt Sử Kí Toàn Thư là bộ có nhiều Page nhất với 933 Page, đây cũng một Bản ghi lại nhiều trang sử vẻ vang nhất của dân tộc, từ sự khởi đầu của Nghìn năm Văn hiến, theo sau với 3 lần chiến thắng quân Mông Nguyên, cho tới Khởi nghĩa Lam Sơn toát lên ý chí quật cường của một quốc gia độc lập.

Bảng 5.1. Thống kê dữ liệu thô sau khi thu thập

Tên tác phẩm	Số lượng page
Lục Vân Tiên	104
Truyện Kiều bản 1866	100
Truyện Kiều bản 1871	136
Truyện Kiều bản 1872	163
ĐVSKTT Quyền Thủ	107
ĐVSKTT Ngoại ký toàn thư	178
ĐVSKTT Bản ký toàn thư	933
ĐVSKTT Bản ký thực lục	787
ĐVSKTT Bản ký tục biên	448

5.3. Quy trình gán nhãn

Như đã đề cập trong 4.1.4, do chi phí gán nhãn quá lớn nên chúng tôi sẽ không thực hiện gán theo mức ký tự. Đối với các Page chữ Hán-Nôm, mỗi cột trong ảnh có thể tồn tại nhiều bounding box (được bọc bởi khung màu đỏ trong ảnh dưới).



Hình 5.5. Minh họa một cột có thể gồm nhiều bounding box

Vì vậy, chúng tôi sẽ xem từng bounding box như một từ và xử lý chúng theo word-level, chứ không phải một cột và xử lý chúng theo line-level. Ngoài ra còn một lợi ích khác cho cách làm này là ngữ cảnh xung quanh một chữ Hán-Nôm sẽ không bị đánh mất. Từ đó, có thể giúp chúng tôi triển khai các mô hình Recognition không chỉ có thể nhận dạng các chữ Hán-Nôm mà còn học được cả ngữ cảnh quanh chúng.

5.3.1. Xây dựng hướng dẫn (Guideline)

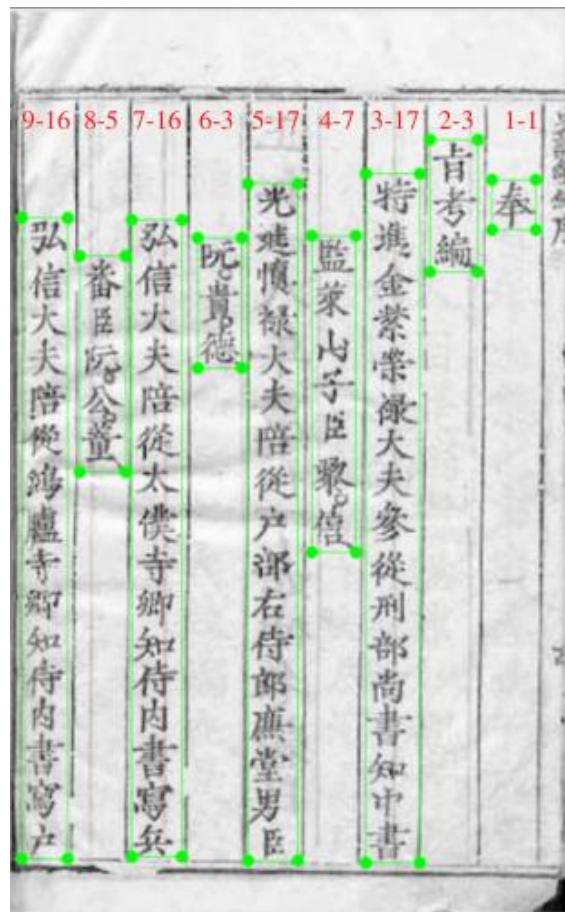
Bộ dữ liệu NomNaOCR được xây dựng với mục đích giải quyết 2 bài toán chính của OCR dành cho chữ Hán-Nôm gồm: *Text Detection* – Phát hiện các vùng ảnh (Patch) chứa các ký tự Hán-Nôm và *Text Recognition* – Nhận dạng các ký tự Hán-Nôm trong các vùng ảnh đó. Vì vậy, chúng tôi sẽ đặt ra 2 bộ qui tắc gán nhãn (Guideline) riêng cho mỗi bài toán.

Với *Text Detection*, mục đích bài toán này là xác định bounding box cho các vùng ảnh chứa câu Hán-Nôm trong mỗi Page. Vì cách viết người xưa là theo chiều đọc nên chúng tôi yêu cầu các bounding box cũng phải được gán theo chiều đọc như vậy. Bên cạnh đó, khác với các bài toán *Object Detection*, các văn bản được viết tách biệt với nhau nên không xảy ra các trường hợp là trong một vùng ảnh xuất hiện nhiều văn bản chồng chéo lên nhau nên các box này không được nằm trùng nhau. Yêu cầu cuối cùng là đảm bảo văn bản cần được nằm trọn vẹn trong một box.

Với *Text Recognition*, mục đích bài toán này là nhận dạng các ký tự Hán-Nôm trong các vùng ảnh nói trên hay các Patch. Có một khó khăn là chúng tôi không thành thạo hệ thống chữ Hán-Nôm nên việc đánh máy trực tiếp các ký tự kỹ thuật số này là không thể hay kể cả việc có thể sao chép trực tiếp từng phần dịch một tương ứng trên website cũng sẽ rất dễ mắc sai lầm. Vì vậy, việc ánh xạ (map) thủ công các ký tự này vào từng Patch là vô cùng tốn kém và không khả thi.

Do đó với bài toán *Text Recognition*, chúng tôi sẽ thực hiện gán nhãn bán thủ công bằng cách đánh số thứ tự của từng Patch theo một quy luật nhất định là từ trên xuống dưới và từ phải sang trái, quy luật này dựa vào cách viết của các tác phẩm Hán-Nôm, theo sau là số lượng ký tự trong từng Patch được ngăn cách bởi dấu “-”.

Như vậy cấu trúc gán nhãn sẽ là là $O-C$ với O là số thứ tự và C là số lượng ký tự. Cuối cùng chúng tôi viết chương trình so khớp các ký tự trong phần dịch của website với từng Patch dựa theo thứ tự O và số lượng C gán được. Phương pháp này không những giúp chúng tôi tối ưu được chi phí xây dựng bộ dữ liệu mà còn tận dụng được các phần dịch được tạo sẵn bởi VNPF.



Hình 5.6. Cách gán nhãn với vị trí và số lượng ký tự cho Patch (màu đỏ)

5.3.2. Gán nhãn tự động (Auto annotation)

Như đã đề cập ở trên, chúng tôi xây dựng bộ dữ liệu NomNaOCR theo phương pháp bán thủ công nên để thu được các Patch trong ảnh, trước tiên chúng tôi sử dụng công cụ PPOCRLLabel thuộc hệ sinh thái của PaddleOCR, đã đề cập trong 4.1.6, để gán tự động các bounding box. Công cụ này mặc định sử dụng DBNet [56] để phát hiện văn bản, đây cũng là mô hình chúng tôi sẽ thử nghiệm cho bài toán Text Detection của mình.

Tuy nhiên, với các ảnh trong dữ liệu của chúng tôi thì công cụ này đa phần sẽ phát hiện các vùng ảnh chứa văn bản trong chúng theo chiều ngang nên chúng tôi sẽ thực hiện quay ảnh theo các góc 90 độ để phù hợp với bài toán. Như vậy, tùy vào từng tác phẩm mà chúng tôi sẽ chọn xoay ảnh theo một trong hai hướng là +90 hoặc -90 độ, sau đó đưa ảnh vào PPOCRLLabel để dự đoán các bounding box.

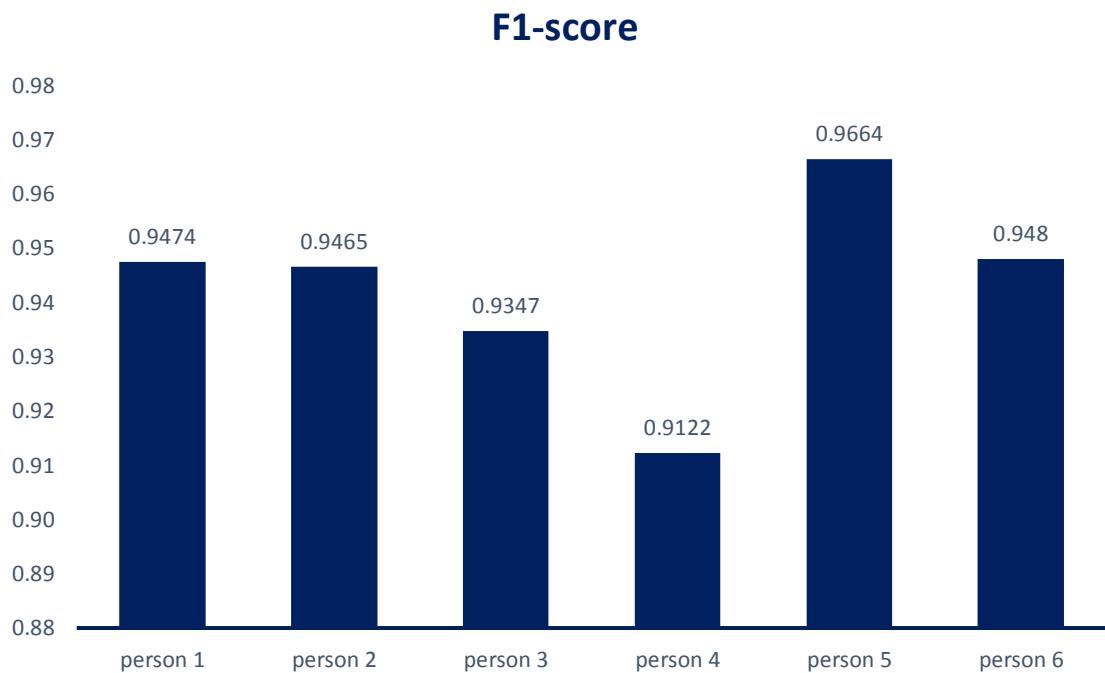
Bên cạnh đó, có những tác phẩm mà chúng tôi phải kết hợp quay cả 2 hướng để có được các bounding box tốt nhất có thể. Đối với cách quay này, chúng tôi sẽ xử lý việc các bounding box ở 2 hướng bị chồng chéo lên nhau bằng cách sử dụng thuật toán Non-maximum Suppression (NMS) để loại bỏ các box dư thừa và giữ lại box tốt nhất. Cuối cùng, chúng tôi triển khai cho các annotator kiểm tra và chỉnh sửa lại các bounding box chưa được đúng để thỏa các yêu cầu đặt ra trong mục [5.3.1](#).

5.3.3. Quy trình đánh giá

Việc đánh giá Guideline và đào tạo cho các annotator là công việc vô cùng quan trọng, vì đó là chìa khóa quyết định bộ dữ liệu sau khi hoàn thành có tốt hay không. Do đó, để đánh giá các annotator sau khi được đào tạo chúng tôi tiến hành lấy ngẫu nhiên 100 ảnh đã được gán tự động trong bộ dữ liệu để cung cấp cho các annotator kiểm tra bounding box và gán nhãn vị trí cũng như là số lượng ký tự cho các bounding box đó.

Đồng thời chúng tôi cũng xây dựng tập dữ liệu đánh giá từ 100 ảnh trên theo nguyên tắc đã đề ra trong [5.3.1](#). Cuối cùng, kết quả sẽ được tổng hợp lại để đánh giá các annotator có đạt yêu cầu huấn luyện hay không bằng cách tính toán kết hợp F1-score cho các bounding box đã được kiểm tra cùng các nhãn của nó, dựa trên một cách tính mới hiệu quả hơn có tên là CLEval [57], được trình bày trong [8.1.1](#).

Do nhãn cho bộ dữ liệu này chỉ cần xác định đúng patch, thứ tự và số lượng ký tự của nó. Nên các annotator đạt được kết quả khá cao, cụ thể, người thấp nhất đạt được 0.9122 còn người cao nhất là 0.9664, kết quả chi tiết được thể hiện qua biểu đồ bên dưới. Vì kết quả của các annotator vượt ngưỡng tốt là 0.8 nên chúng tôi sẽ triển khai cho kiểm tra bounding box và gán nhãn độc lập.



Hình 5.7. Kết quả đánh giá các annotator

5.3.4. Triển khai thực tế

Trong khi triển khai kiểm tra các bounding box sau khi được gán tự động và phân công gán nhãn độc lập cho các annotator, chúng tôi cũng tổ chức thêm một đội để kiểm tra các ký tự Hán-Nôm kỹ thuật số có khớp và chính xác với ảnh trên trang website của VNPF không? Nếu không, chúng tôi sẽ tiến hành xử lý như sau: với các ký tự sai hoặc thiếu so với ảnh chúng tôi thay thế hoặc thêm các ký tự “?” vào vị trí bị thiếu sót, còn đối với các ký tự dư so với ảnh, chúng tôi xóa ký tự đó. Mục đích của việc kiểm tra và chỉnh sửa này là nhằm đảm bảo các ký tự Hán-Nôm chính xác hoàn toàn với ảnh để có được nhãn tốt nhất cho giai đoạn gắn (map) nhãn tự động.

Tiếp đến, chúng tôi giao cho những người xây dựng Guideline nhiệm vụ là kiểm tra lại các bounding box cùng với nhãn vị trí và số lượng các ký tự Hán-Nôm mà những annotator đã gán có thỏa mãn các yêu cầu trong 5.3.1 chưa, nếu chưa chúng tôi sẽ chỉnh sửa lại cho phù hợp với các tiêu chí đã đặt ra.

Cuối cùng, chúng tôi sẽ chạy chương trình để map tự động các nhãn là các ký tự Hán-Nôm kỹ thuật số với các bounding box dựa vào vị trí và số lượng các ký tự

tương ứng trong chúng. Nhưng một vấn đề nảy sinh trong giai đoạn này là tổng số lượng các ký tự đã được đếm ở mỗi bounding box có thể không khớp với tổng số ký tự thực tế có trong ảnh.

Để giải quyết vấn đề trên, chúng tôi tiến hành kiểm tra lại số lượng ký tự trong mỗi bounding box có khớp với số lượng ký tự xuất hiện trong bản cắt ảnh hay Patch mà bounding box đó đã bao phủ hay không. Nếu số lượng ký tự cho bounding box sai thì chúng tôi sẽ tiến hành chỉnh sửa lại giá trị đó. Nếu tại Patch đó xuất hiện ký tự dư thừa hoặc bị thiếu ký tự nào đó do trong bước kiểm tra từng ký tự kỹ thuật số so với ảnh có xảy ra sai sót thì chúng tôi sẽ thực hiện thêm dấu “?” vào vị trí bị thiếu hoặc xóa ký tự dư thừa. Số lượng ảnh không khớp theo từng tác phẩm được thể hiện qua bảng dưới.

Bảng 5.2. Thông kê các page lệch nhau giữa website của VNPF và annotator

Tên tác phẩm	Số lượng các page bị lệch
Lục Vân Tiên	3
Truyện Kiều bản 1866	3
Truyện Kiều bản 1871	1
Truyện Kiều bản 1872	2
ĐVSKTT Quyền Thủ	36
ĐVSKTT Ngoại ký toàn thư	48
ĐVSKTT Bản ký toàn thư	242
ĐVSKTT Bản ký thực lục	124
ĐVSKTT Bản ký tục biên	154

Từ bảng trên có thể thấy, các tác phẩm thơ có số lượng ảnh không khớp rất ít vì các tập thơ này được viết theo thể thơ lục bát với các câu ở trên là 6 ký tự và các câu ở dưới là 8 ký tự ([hình 5.3](#)) nên có cấu trúc rõ ràng và khá đẹp, còn số lượng ảnh không khớp của bộ Đại Việt Sử Ký Toàn Thư nói chung thì khá nhiều vì đây có thể xem là một tác phẩm văn xuôi nên thường sẽ có cấu trúc không được rõ ràng và đồng nhất. ([hình 5.4](#)).

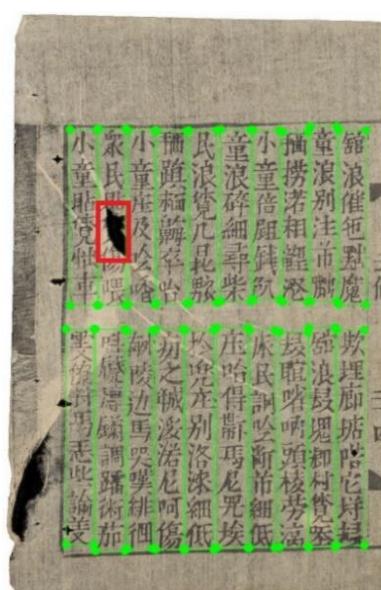
5.4. Các khó khăn cùng hướng xử lý

Trong quá trình xây dựng bộ dữ liệu NomNaOCR, chúng tôi đối mặt với nhiều khó khăn vì bộ dữ liệu này được xây dựng từ dữ liệu thô là các tác phẩm Hán-Nôm viết tay cách đây hàng trăm năm. Do đó, tình trạng giấy khi được lưu trữ đến hiện nay của một số tác phẩm không được tốt. Bên cạnh đó, phần dịch bằng các ký tự Hán-Nôm kỹ thuật số của VNPF cũng xảy ra các lỗi về nội dung và chính tả. Cách giải quyết chung của chúng tôi khi các Patch đạt yêu cầu ở bài toán phát hiện vùng ảnh chứa các ký tự Hán-Nôm nhưng lại không đạt ở bài toán nhận dạng các ký tự đó thì chúng tôi sẽ đánh dấu các Patch này là “difficult”, để khi tiến hành cắt Patch từ ảnh gốc làm đầu vào cho bài toán nhận dạng, chương trình sẽ bỏ qua các Patch được đánh dấu là “difficult”. Sau đây là các khó khăn, lỗi, cùng hướng giải quyết cụ thể:

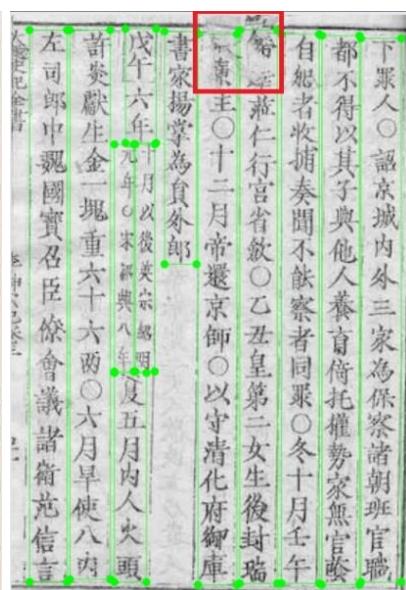
- Bộ dữ liệu xuất hiện các ảnh bị quá mờ, bị vết mực hoặc vết rách lớn trong các Patch, dẫn đến không thể nhận diện được các ký tự. Chúng tôi giải quyết bằng cách đánh dấu các Patch đó là “difficult”.



Hình 5.8. Ảnh vết mực

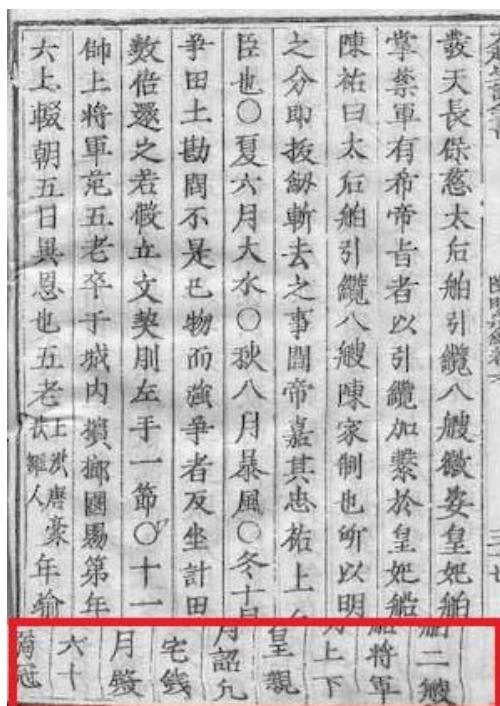


Hình 5.9. Ảnh vết rách



Hình 5.10. Ảnh chữ mờ

- Bộ dữ liệu cũng tồn tại các Page mà không chứa bất kỳ ký tự nào, cũng như các Page bị đè lên nhau khi quét từ dạng giấy sang dạng hình ảnh. Chúng tôi giải quyết bằng cách xóa và không gán nhãn các Page này.



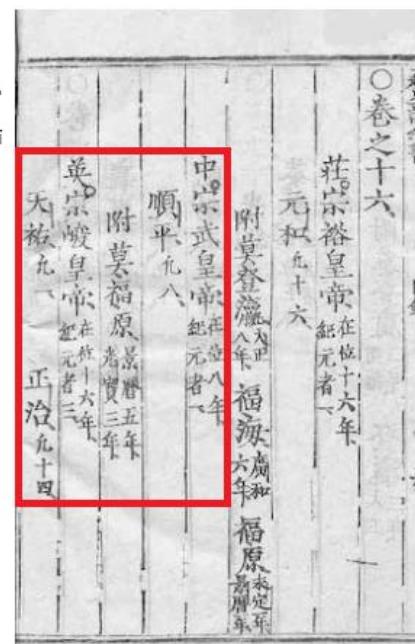
Hình 5.11. Ảnh bị gấp đè



Hình 5.12. Ảnh trống

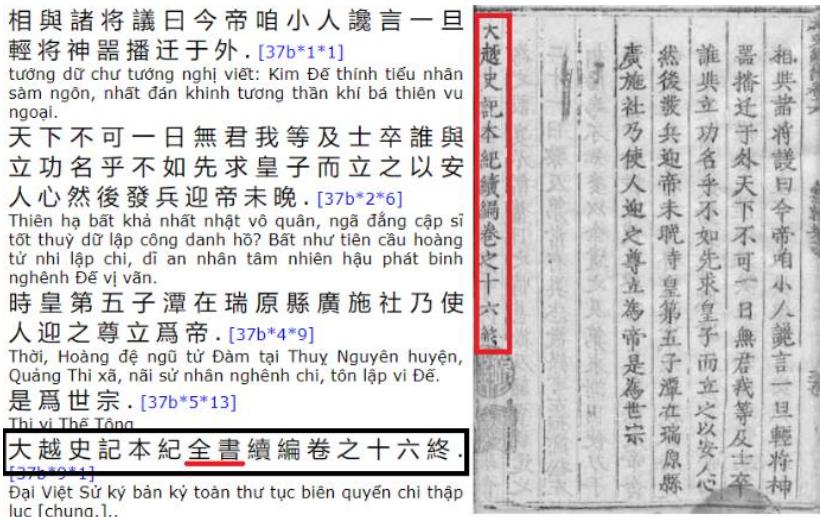
- Trong giai đoạn kiểm tra các ký tự được phiên dịch có chính xác với hình ảnh trên website của VNPF không, chúng tôi phát hiện một số ký tự trong ảnh không được dịch lại hoặc bị chú thích sai. Cách giải quyết cho các trường hợp này là thêm dấu “?” vào những chỗ thiếu hoặc sai.

○ 卷之十六. [9b*01*02]
 Quyển chi thập lục
 莊宗裕帝在位十六年紀元者一. [9b*02*01]
 Trang Tông Dụ Hoàng Đế [tại vị thập lục niên ki
 nguyên già nhất].
 元和凡十六. [9b*03*01]
 Nguyễn Hoà [phàm thập lục].
 附莫登瀛大正八年. [9b*04*01]
 Mac Đăng Doanh [Đại Chính bát niên].
 福海廣和六年. [9b*04*09]
 Phúc Hải [Quảng Hoà lục niên].
 福源永定一年景曆一年. [9b*04*15]
 Phúc Nguyên [Vĩnh Định nhâ



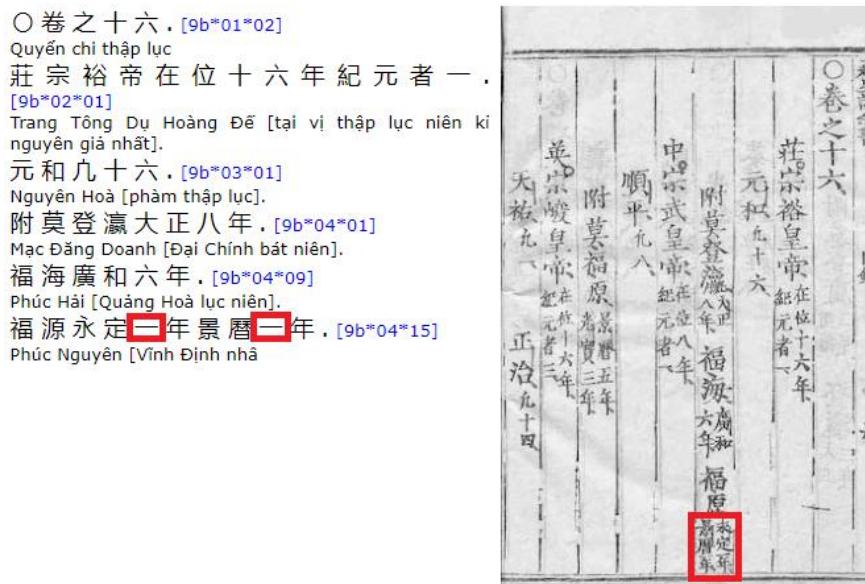
Hình 5.13. Ảnh bị dịch thiếu (vùng khoanh màu đỏ không có phần dịch)

- Ngoài ra, chúng tôi cũng phát hiện một số ký tự trong phần dịch bị dư so với ảnh, nên chúng tôi quyết định xóa đi các ký tự dư này.



Hình 5.14. Ảnh bị dịch dư

- Trong giai đoạn so khớp tự động các nhãn văn bản được dịch với các Patch đã được phát hiện, có xuất hiện các trường hợp một số ký tự trong ảnh dễ gây nhầm lẫn vì chúng khá khó thấy. Như trong hình dưới có ký tự “—” rất khó nhìn thấy trong ảnh, dẫn đến các annotator bỏ sót ký tự này. Tuy nhiên, khi kiểm tra lại thì ký tự này có trong ảnh và tồn tại trong phần dịch của website nên chúng tôi quyết định thêm ký tự này vào nhãn.



Hình 5.15. Ảnh có ký tự khó để thấy

5.5. Phân tích và chia dữ liệu

Sau khâu triển khai thực tế, bộ dữ liệu NomNaOCR được xử lý và thu được 2953 Page từ các tác phẩm: Lục Vân Tiên, Truyện Kiều (các bản năm 1866, 1871, 1872), và trọn bộ Đại Việt Sử Ký Toàn Thư. Bằng cách gán nhãn bán thủ công cho các tác phẩm trên, chúng tôi có được thêm 38318 Patch. Ngoài các tác phẩm thơ chiếm khoảng 17,03% Page trong bộ dữ liệu, thì phần còn lại là bộ Đại Việt Sử Ký Toàn Thư chiếm khoảng 82,97%, nên các câu chủ yếu có độ dài 6, 8, 17 và 18 ký tự.

Bảng 5.3. Số lượng câu theo độ dài

Chiều dài câu	Số lượng câu
1	631
2	1318
3	1371
4	1232
5	727
6	6023
7	406
8	5824
9	396
10	432
11	436
12	519
13	331
14	290
15	254
16	588
17	2679
18	10506
> 18	4355

Bảng 5.4. Tần suất ở cấp độ ký tự của bộ dữ liệu NomNaOCR

Khoảng tần suất	Số ký tự khác nhau	Tổng số lần xuất hiện
1	1621	1621
2-5	1875	5673
6-10	856	6560
11-20	766	11291
21-50	894	29561
51-100	567	40512
> 100	930	364374
Tổng cộng	7509	459547

Tổng số lượng các ký tự khác nhau hay số ký tự tập từ vựng (vocab) trong bộ dữ liệu là 7509 với tần suất xuất hiện được mô tả như bảng trên. Có thể thấy rằng trong tập từ vựng, một số lớn các ký tự xuất hiện chỉ một lần và chúng chiếm một phần không nhỏ lên đến 21,59% số từ vựng nhưng lại có tần suất xuất hiện chỉ 0.35% trong toàn bộ dữ liệu. Kết quả trên không có gì ngạc nhiên vì với hệ thống chữ Hán-Nôm được viết trong khoảng thời gian dài lịch sử, mỗi ký tự có thể chỉ đại diện riêng cho một sự vật, sự việc hay tên người, địa danh, ... nên có tần suất xuất hiện rất nhỏ nhưng vẫn chiếm một số lượng khá lớn trong tập từ vựng của bộ dữ liệu.

Theo quan điểm trong huấn luyện học sâu thì rõ ràng là chúng tôi cần tính đến đặc điểm này vì mô hình học sâu không thể tự dự đoán một nhân vật, một địa điểm, hoặc một ký tự bất kỳ mà nó chưa từng được học từ trước. Do đó khi thực hiện việc chia dữ liệu thành các tập huấn luyện (Train) và tập kiểm tra (Validate), số lượng các điểm dữ liệu chứa các ký tự chỉ xuất hiện một lần trong tập Validate nên càng ít càng tốt. Vì vậy, để việc chia dữ liệu đạt được yêu cầu, chúng tôi sẽ xác định một giá trị R cho mỗi điểm dữ liệu s trong tập dữ liệu D cùng N là tổng tần suất của mỗi ký tự trong s xuất hiện trong D/s [9]. Điểm R được tính toán theo công thức (5.1):

$$R_s = N_{distinct(s)} \times \max_i^D N_i + N_s \quad (5.1)$$

Như được thể hiện trong công thức trên, thứ nhất điểm R nhấn mạnh vào số ký tự trong s xuất hiện trong D, và thứ hai là nhấn mạnh vào tần suất xuất hiện của những ký tự đó. Để chọn k mẫu cho tập Validate, chúng tôi chỉ cần lấy k mẫu có điểm R cao nhất. Ở đây, chúng tôi sẽ chọn k = 7664 tương đương với 20% của bộ dữ liệu. [Bảng 5.4](#) cho thấy tần suất các điểm dữ liệu và sự giao nhau của các ký tự giữa 2 tập Train và Validate.

[Bảng 5.5. Thống kê giao nhau giữa 2 tập Train và Validate trong NomNaOCR](#)

Tập dữ liệu	Số điểm dữ liệu	Tỉ lệ ký tự giao nhau
Tập Train	30654	93.24%
Tập Validate	7664	64.41%

Thuật ngữ giao nhau giữa các ký tự để cập đến phạm vi bao phủ về từ vựng của tập dữ liệu này so với tập dữ liệu kia. Từ [bảng 5.4](#), có thể thấy bảng cách áp dụng điểm R để chia dữ liệu, chúng tôi có được 93.24% các ký tự khác nhau trong tập Validate tồn tại trong tập Train, nghĩa là tập Train bao gồm hầu hết mọi ký tự trong Validate.

Bảng 5.6. Thống kê các Patch trong tập Train và Validate theo từng tác phẩm

Tập dữ liệu	Tên tác phẩm	Patch
Tập Train	Lục Vân Tiên	1646
	Truyện Kiều bản 1866	1902
	Truyện Kiều bản 1871	2609
	Truyện Kiều bản 1872	2552
	ĐVSKTT Quyển Thủ	737
	ĐVSKTT Ngoại ký toàn thư	2065
	ĐVSKTT Bản ký toàn thư	8923
	ĐVSKTT Bản ký thực lục	6349
	ĐVSKTT Bản ký tục biên	3871
Tập Validate	Lục Vân Tiên	407
	Truyện Kiều bản 1866	483
	Truyện Kiều bản 1871	639
	Truyện Kiều bản 1872	702
	ĐVSKTT Quyển Thủ	192
	ĐVSKTT Ngoại ký toàn thư	540
	ĐVSKTT Bản ký toàn thư	2195
	ĐVSKTT Bản ký thực lục	1542
	ĐVSKTT Bản ký tục biên	964

Còn đối với phần dữ liệu để giải quyết bài toán phát hiện vùng ảnh chứa văn bản Hán-Nôm, chúng tôi sẽ lấy ngẫu nhiên 20% dữ liệu trên từng tác phẩm làm tập Validate và 80% còn lại của bộ dữ liệu trên mỗi tác phẩm để làm tập Train. Do mỗi tác phẩm có điều kiện ảnh như màu sắc, phông nền, ánh sáng, ... khác nhau, vì vậy

mục đích cho cách chia dữ liệu như trên là để tránh trường hợp tập Train bị thiếu đi các điều kiện ảnh có trong tập Validate, ngoài ra, còn để giúp mô hình có thể học được các dạng tổng quát của từng tác phẩm.

Bảng 5.7. Thông kê các Page trong tập Train và Validate theo từng tác phẩm

Tập dữ liệu	Tên tác phẩm	Page
Tập Train	Lục Vân Tiên	83
	Truyện Kiều bản 1866	80
	Truyện Kiều bản 1871	108
	Truyện Kiều bản 1872	130
	ĐVSKTT Quyển Thủ	84
	ĐVSKTT Ngoại ký toàn thư	142
	ĐVSKTT Bản ký toàn thư	745
	ĐVSKTT Bản ký thực lục	629
	ĐVSKTT Bản ký tục biên	358
Tập Validate	Lục Vân Tiên	21
	Truyện Kiều bản 1866	20
	Truyện Kiều bản 1871	28
	Truyện Kiều bản 1872	33
	ĐVSKTT Quyển Thủ	21
	ĐVSKTT Ngoại ký toàn thư	36
	ĐVSKTT Bản ký toàn thư	187
	ĐVSKTT Bản ký thực lục	158
	ĐVSKTT Bản ký tục biên	90

5.6. Bộ dữ liệu Synthetic Nom String

Hiện nay, phương pháp huấn luyện mô hình trước (Pre-training) với một bộ dữ liệu lớn, sau đó tinh chỉnh lại (Fine-tuning) với bộ dữ liệu cụ thể nhỏ hơn có thể sẽ mang lại nhiều kết quả tốt và giảm được chi phí cho việc phải thu thập thêm nhiều dữ liệu.

Vì thế, chúng tôi sẽ sử dụng một bộ dữ liệu lớn, có sẵn dành cho chữ Hán-Nôm để thực hiện tiền huấn luyện cho các mô hình của bài toán nhận dạng các ký tự Hán-Nôm trong Patch, sau đó tinh chỉnh lại trên bộ dữ liệu NomNaOCR của chúng tôi.

Bộ dữ liệu lớn nói trên được chúng tôi sử dụng ở đây là Synthetic Nom String, một phần của bộ dữ liệu IHR-NomDB [9], bao gồm 101621 ảnh là các Patch được tạo tự động từ ngân hàng các câu Hán-Nôm được thu thập bởi tác giả Vu Manh Tu và các cộng sự. Các câu Hán-Nôm trong bộ dữ liệu này có số lượng ký tự dao động từ 1 đến 10 ký tự nhưng số lượng các ký tự trong mỗi câu hầu hết từ 6 đến 8 ký tự.

Bảng 5.8. Thống kê số lượng câu theo độ dài trong Synthetic Nom String.

Chiều dài câu	1	2	3	4	5	6	7	8	9	10
Số lượng câu	253	1652	1030	2087	7043	25631	31984	17041	8221	6679

Tổng cộng có 13177 ký tự Hán-Nôm khác nhau xuất hiện trong bộ dữ liệu này, phân bố trong các khoảng tần suất xuất hiện khác nhau, tương tự như bộ dữ liệu NomNaOCR chúng tôi xây dựng, bộ dữ liệu này cũng có một lượng lớn các ký tự chỉ xuất hiện một lần, khoảng 21,61% trong tập từ vựng. Vì thế, tác giả đã áp dụng công thức do chính tác giả đã đề xuất để chia bộ dữ liệu này thành 2 tập huấn luyện và kiểm tra, đã được mô tả chi tiết trong [5.5](#).

Bảng 5.9. Tần suất ở cấp độ ký tự của bộ dữ liệu Synthetic Nom String

Khoảng tần suất	Số ký tự khác nhau	Tổng số lần xuất hiện
1	2847	2847
2-5	4061	12446
6-10	1512	11421
11-20	1211	17719
21-50	1293	42325
51-100	843	60023
> 100	1410	558210
Tổng cộng	13177	704991

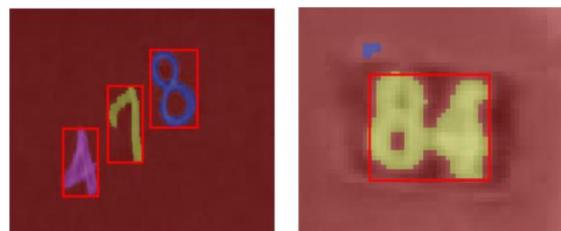
Chương 6. CÁC PHƯƠNG PHÁP TIẾP CẬN

6.1. Khởi nguồn và lý do tiếp cận bằng Học sâu

Như đã đề cập trong 4.1.3, một trong những hướng tiếp cận đơn giản nhất cho bài toán OCR là sử dụng các kỹ thuật thị giác máy tính truyền thống hay các phương pháp xử lý ảnh căn bản và chúng đã được sử dụng trong một thời gian dài. Có thể gói gọn chúng trong các bước sau:

- Áp dụng các bộ lọc (filters) để làm nổi bật các ký tự từ background.
- Tiến hành phát hiện cạnh (contour) để phát hiện các vùng ảnh chứa ký tự.
- Thực hiện phân loại cho các vùng ảnh đó để xác định ký tự trong chúng là gì.

Rõ ràng là nếu bước 2 được thực hiện tốt thì bước 3 có thể dễ dàng thực hiện bằng cách khớp mẫu (Pattern Matching) hay huấn luyện một mạng CNN đơn giản. Tuy nhiên, phát hiện cạnh sẽ khá khăn cho việc khai quát hóa, chưa thể thích ứng được với các sự đa dạng trong ảnh, nhất là trong các môi trường hỗn tạp. Nó đòi hỏi rất nhiều điều chỉnh thủ công nên cách làm này trở nên không khả thi trong hầu hết các trường hợp [17].



Hình 6.1. Xử lý ảnh căn bản thất bại khi các ký tự gần nhau (Nguồn [17])

Một nhược điểm nữa cũng có thể thấy là đa phần cách làm này chỉ có hiệu quả với mức ký tự nên nếu chúng tôi triển khai để tài theo hướng này thì ngoài chi phí gán nhãn lớn khi làm dữ liệu, chúng tôi còn sẽ đánh mất ngữ cảnh xung quanh một chữ Hán-Nôm như đã đề cập trong 5.3. Các hướng tiếp cận bằng Deep Learning hiện nay có thể xem là đã chín mùi, vượt trội trong việc khai quát của chúng và đã trở thành giải pháp thống trị trong cả nghiên cứu và thực tế, có thể giúp chúng tôi đạt được kết quả tốt cho đề tài này.

6.2. Phát hiện văn bản (Text Detection)

Đầu tiên, có thể thấy bài toán phát hiện văn bản khá giống với bài toán phát hiện vật thể (Object Detection) trong đó object cần được phát hiện chính là văn bản. Do vậy, ta có thể áp dụng các kiến trúc Deep Learning cho bài toán này như YOLO, SSD hay Faster R-CNN để đạt được độ chính xác cao hơn so với các phương pháp xử lý ảnh thủ công.

Tuy nhiên, các mô hình này có vẻ chỉ mang lại kết quả tốt với các object lớn cũng như với ảnh có độ phân giải cao [58] – những thứ xa xỉ đối với đa phần dữ liệu của chúng tôi. Khá là trớ trêu vì trên thực tế, các mô hình này tỏ ra khó khăn hơn và có xu hướng không đạt được độ chính xác mong muốn khi detect các chữ số và chữ cái so với khi detect các object phức tạp như chó, mèo hay con người.

Các phương pháp chuyên biệt dựa trên Deep Learning gần đây đã giải quyết được hầu hết các vấn đề trên và đạt được những kết quả tốt trên nhiều bộ dữ liệu benchmark tiêu chuẩn như ICDAR 2013 [19] và 2015 [20] hay Total-Text [21]. Đây đồng thời cũng là những hướng tiếp cận mà chúng tôi sẽ sử dụng cho bài toán Text Detection của đề tài này. Cụ thể, chúng được chia thành 2 loại chính như sau:

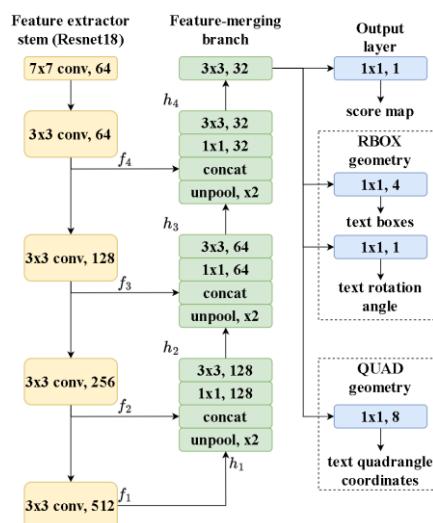
- Các phương pháp Regression-based: sau khi có kết quả dự đoán, bounding box cuối cùng sẽ được lọc qua bằng thuật toán NMS. Phần lớn box của cách làm này chỉ có 4 điểm tọa độ nên phương pháp này sẽ bị giới hạn trong việc biểu diễn các văn bản có hình dạng bất thường (ví dụ như bị cong) trong ảnh. Một số mô hình nổi bật cho phương pháp này có thể kể đến như EAST hay TextBoxes.
- Các phương pháp Segmentation-based: xem bài toán Detection dưới góc nhìn của một bài toán phân vùng vật thể (Object Segmentation), nhằm mục đích tìm kiếm các vùng ảnh chứa văn bản ở cấp pixel (trả về xác suất một pixel chứa văn bản). Cách tiếp cận này sẽ phát hiện văn bản bằng cách ước tính vùng giới hạn (bounding area) của chữ. Bounding box của văn bản sẽ được xây dựng thông qua kết quả phân vùng, nhờ vậy chúng có thể biểu diễn các hình dạng bất thường. Một số đại diện tiêu biểu cho phương pháp này là DBNet, PSENet, ...

- Ngoài 2 loại chính trên, còn một số phương pháp khác như Character-based methods (nổi tiếng với CRAFT), Word-based methods, Tuy nhiên, chúng chỉ là cách phân loại cho một số giải pháp/model cụ thể nên ta sẽ không đi sâu vào.

6.2.1. Tiếp cận theo Regression-based với EAST

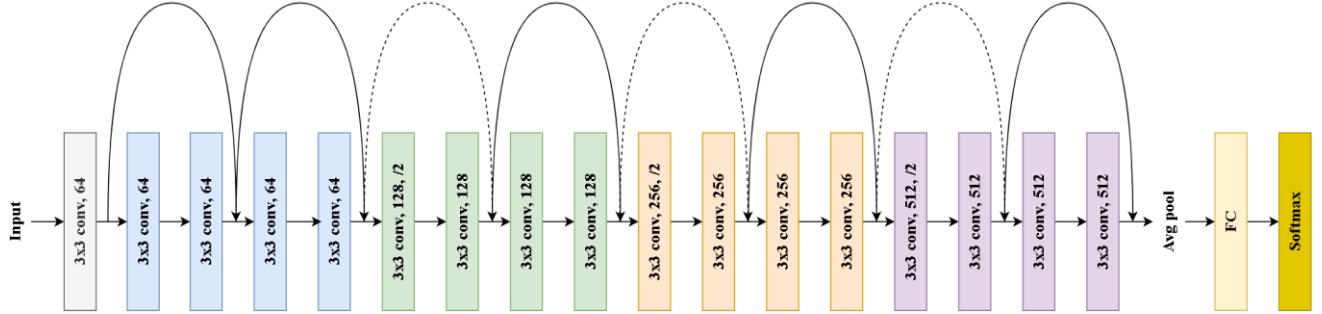
EAST (*Efficient and Accurate Scene Text Detector*) [59] sử dụng một mạng nơ-ron duy nhất để dự đoán văn bản ở mức word-level hoặc line-level. Nó có thể phát hiện văn bản theo mọi hướng tùy vào các hình dạng tứ giác của bounding box. Vào năm 2017, mô hình này vượt trội hơn hẳn các phương pháp hiện đại lúc đó. Mô hình này bao gồm một mạng tích chập đầy đủ đóng vai trò chính phát hiện văn bản trong ảnh, kết hợp với NMS để đưa ra các dự đoán bằng cách hợp nhất nhiều bounding box được phát hiện nhưng chưa đúng thành một bounding box chính xác.

Kiến trúc của EAST được tạo ra để xem xét các kích thước khác nhau của các Patch. Ý tưởng được đề ra là việc phát hiện các Patch có kích thước lớn sẽ yêu cầu các đặc trưng được trích xuất từ giai đoạn sau của mạng, trong khi phát hiện các Patch có nhỏ hơn thì cần đến các đặc trưng ở giai đoạn đầu của mô hình. Vì vậy, tác giả của EAST đã đề xuất một kiến trúc có 3 nhánh hay 3 phần để kết hợp thành một mạng nơ-ron, bao gồm: Trích xuất đặc trưng trong ảnh (Feature Extractor Stem), Hợp nhất các đặc trưng đó (Feature Merging Branch) và cuối cùng là lớp đầu ra.



Hình 6.2. Kiến trúc mô hình EAST

Với nhánh Trích xuất đặc trưng, các kiến trúc mạng được sử dụng trong phần này có thể là bất kỳ mạng tích chập nào như là PVANet hay VGG16. Ở đây, kiến trúc mạng chúng tôi sẽ sử dụng là ResNet 18, từ đó thu được các feature map tương ứng là f_1, f_2, f_3 và f_4 từ các đầu ra của mô hình là conv2, conv3, conv4, và conv5.



Hình 6.3. Kiến trúc của ResNet 18

Nhánh Hợp nhất đặc trưng sẽ có nhiệm vụ hợp nhất các đầu ra đặc trưng từ các lớp khác nhau của mạng ResNet 18 với đầu vào là các hình ảnh được đưa qua mô hình Resnet18 và các đầu ra là 4 feature map nói trên. Việc hợp nhất các feature map sẽ có chi phí tính toán cao. Đó là lý do tại sao EAST sử dụng kiến trúc U-Net, được đề cập trong 4.2.2.3, để hợp nhất dần dần các feature map. Đầu tiên, các đặc trưng thu được từ lớp conv5 sẽ được đưa vào lớp UnPooling để tăng gấp đôi kích thước. Lúc này, kích thước các đặc trưng sẽ bằng với kết quả đầu ra từ lớp conv4 và cả 2 lớp sau đó sẽ được hợp nhất thành một lớp. Sau đó, một lớp tích chập với kích thước 1×1 được sử dụng để giảm số lượng kênh và số lượng tính toán, tiếp đến là một lớp tích chập 3×3 được dùng để tổng hợp thông tin. Quá trình trên diễn ra tương tự cho các lớp đầu ra khác của mô hình ResNet 18. Một lớp tích chập 3×3 cuối cùng sẽ được áp dụng để tạo ra feature map cho giai đoạn Hợp nhất này trước khi được đưa vào nhánh đầu ra. Các tính toán cho quá trình này, đại diện bởi các khối h_i trong hình 6.2, lần lượt như sau:

$$g_i = \begin{cases} \text{unpool}(h_i) & \text{if } i \leq 3 \\ \text{conv}_{3 \times 3}(h_i) & \text{if } i = 4 \end{cases} \quad (6.1)$$

$$h_i = \begin{cases} f_i & \text{if } i = 1 \\ \text{conv}_{3 \times 3}(\text{conv}_{1 \times 1}([g_{i-1}; f_i])) & \text{otherwise} \end{cases} \quad (6.2)$$

Cuối cùng là lớp đầu ra bao gồm một score map cho biết xác suất trong Patch đó có văn bản hay không và một geometry map để xác định ranh giới của bounding box. Ngoài ra, geometry map có thể là tứ giác (Quadrangle – QUAD) bao gồm 4 tọa độ của một hình chữ nhật hoặc Rotate Box (RBOX), bao gồm tọa độ trên cùng bên trái (top left), chiều rộng (width), chiều cao (height) và góc xoay (angle). Với các mô hình được kế thừa từ các giải pháp trong Object Detection như này, hàm mất mát (loss function) IoU là một trong những hàm mất mát thường được sử dụng. Nhưng ở đây chúng tôi chủ yếu có 2 kết quả đầu ra là score map và geometry map, vì vậy giá trị loss tổng quát cho chúng sẽ được biểu thị qua công thức (6.3):

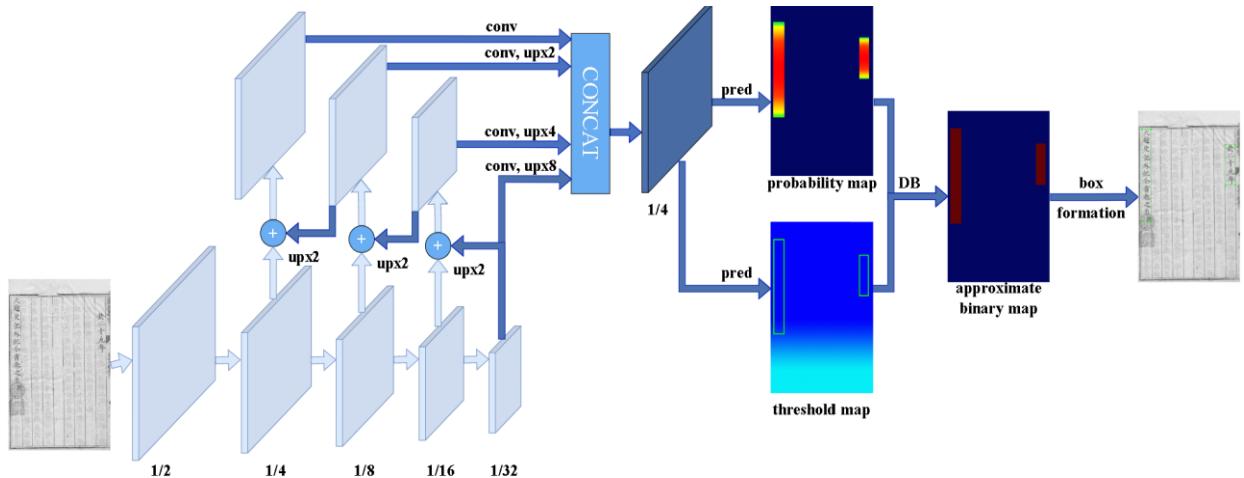
$$L = L_s + \lambda_g L_g \quad (6.3)$$

Trong đó:

- L : Hàm mất mát tổng.
- L_s : Hàm mất mát của score map.
- L_g : Hàm mất mát của score map geometry map.
- λ_g : Mức độ quan trọng cho các loss khác nhau.

6.2.2. Tiếp cận theo Segmentation-based với DBNet

DBNet (*Differentiable Binarization*) [56] sử dụng một mạng tích chập để trích xuất các đặc trưng trong ảnh đầu vào (backbone). Mạng này có thể là VGG16, ResNet 18, ResNet 50, ... Ở đây, chúng tôi cũng sẽ sử dụng Resnet 18 làm backbone tương tự như với EAST để thu được các feature map. Các đặc trưng sẽ được đi qua lớp Un-sample theo cùng một tỷ lệ và xếp tầng để tạo ra feature map F. Tiếp đến, các lớp feature map F được sử dụng để dự đoán cả bản đồ xác suất P (*Probability map*) và bản đồ ngưỡng T (*Threshold map*) từ đó có được bản đồ nhị phân xấp xỉ \hat{B} (*Approximately binary map*). Trong quá trình huấn luyện, việc học giám sát (supervision) sẽ được áp dụng cho 3 map trên, trong đó Probability map và Approximately binary map chia sẻ chung quan sát. Trong giai đoạn dự đoán, các bounding box sẽ thu được từ Approximately binary map hoặc Probability map sau khi đi qua một module tạo box.



Hình 6.4. Kiến trúc mô hình DBNet (Nguồn [56])

Nhị phân hóa (Binarization) tiêu chuẩn là cho trước Probability map $P \in R^{H \times W}$ tạo ra bởi mạng phân vùng, trong đó H và W là chiều dài và rộng của map, sau đó biến đổi nó thành một bản đồ nhị phân $P \in R^{H \times W}$ (Binary map) mà các giá trị điểm ảnh là 1 nếu vùng ảnh chứa văn bản là hợp lệ, quá trình nhị phân hóa như sau:

$$B_{i,j} = \begin{cases} 1 & \text{if } P_{i,j} \geq t, \\ 0 & \text{otherwise,} \end{cases} \text{ với } t \text{ là ngưỡng cho trước, } (i, j) \text{ là tọa độ trong map} \quad (6.4)$$

Vì, việc nhị phân hóa tiêu chuẩn không thể tối ưu cùng với mạng phân vùng trong quá trình huấn luyện. Để giải quyết vấn đề này, tác giả và các cộng sự đã đề xuất thay đổi công thức nhị phân hóa thành công thức (6.5)

$$\hat{B}_{i,j} = \frac{1}{1 + e^{-k(P_{i,j} - T_{i,j})}} \quad (6.5)$$

Trong đó:

- \hat{B} : Approximately binary map.
- T : bản đồ ngưỡng thích ứng (Adaptive threshold map) học được từ mạng
- k : hệ số khuyếch đại (thường là 50).

Nhị phân hóa có thể đạo hàm với ngưỡng thích ứng không chỉ giúp phân biệt các vùng ảnh từ phông nền mà còn các trường hợp văn bản riêng biệt được liên kết chặt chẽ. Với việc sinh nhãn cho một ảnh văn bản, mỗi đa giác của các vùng văn bản được mô tả bởi một tập các phân vùng:

$$G = \{S_k\}_{k=1}^n \quad (6.6)$$

Với n là số lượng đỉnh, có thể khác nhau tùy vào bộ dữ liệu. Sau đó các vùng tích cực (positive area) được sinh bằng cách thu nhỏ các đa giác G đến G_s sử dụng thuật toán Vatti. Độ lệch D của việc thu nhỏ này được tính bằng chu vi L và diện tích A của đa giác ban đầu với r là tỉ lệ thu nhỏ (thường là 0.4):

$$D = \frac{A(1 - r^2)}{L} \quad (6.7)$$

Tương tự, ta có thể sinh các nhãn cho Threshold map. Đầu tiên, đa giác văn bản G được giãn ra với cùng độ lệch D đến G_d . Khoảng cách giữa G_s và G_d sẽ được xem xét như là viền của các vùng văn bản, trong đó nhãn của Threshold map có thể được sinh bằng cách tính toán khoảng cách đến phân vùng gần nhất trong G . Hàm mất mát L có thể được thể hiện như một tổng trọng số của các hàm loss:

$$L = L_s + \alpha \times L_b + \beta \times L_t \quad (6.8)$$

Trong đó, L_s - hàm mất mát cho Probability map và L_b - hàm mất mát cho Binary map, L_t - hàm mất mát cho Threshold map. Dựa trên giá trị của các hàm mất mát này, 2 hệ số α và β sẽ được thêm vào và có giá trị từ 1 đến 10. Theo bài báo [56], một hàm mất mát nhị phân Cross-entropy sẽ được áp dụng cho cả L_s và L_b với S_t là tập được lấy mẫu trong đó tỉ lệ giữa số mẫu tích cực và tiêu cực là 1:3.

$$L_s = L_b = \sum_{i \in S_t} y_i \log x_i + (1 - y_i) \log(1 - x_i) \quad (6.9)$$

L_t được tính toán như là tổng các khoảng cách L1 giữa nhãn dự đoán và nhãn bên trong các đa giác được giãn ra G_d :

$$L_t = \sum_{i \in R_d} |y_i^* - x_i^*| \quad (6.10)$$

Trong đó:

- R_d là tập các vị trí của các điểm ảnh bên trong đa giác được giãn G_d .
- y^* là nhãn cho Threshold map.

6.3. Nhận dạng văn bản (Text Recognition)

Bài toán Text Recognition có thể phân thành 2 loại sau: Nhận dạng chữ thông thường - *Regular Text Recognition* và Nhận dạng chữ có sự biến dạng - *Irregular Text Recognition* (chữ có thể bị nghiêng, cong hoặc bị mờ, méo do nhiều yếu tố). Một số cách làm trước đây là sẽ phát hiện vị trí từng kí tự và nhận dạng chúng bằng quy hoạch động và thuật toán tìm kiếm được đề cập trong bài báo [60]. Các phương pháp này có hạn chế là sẽ không khai thác được mối quan hệ về mặt ngữ nghĩa.

Giả sử, nếu lựa chọn triển khai để tài từ đầu ở mức ký tự thì ở bài toán Recognition này, để biết các bounding box thu được có chứa ký tự gì sau khi thực hiện phát hiện văn bản trên một ảnh, ta chỉ cần đơn thuần áp dụng một kiến trúc CNN. Cách làm này tương tự như bài toán nhận dạng chữ số viết tay trên bộ dữ liệu MNIST [27] kinh điển. Và cũng đã có một cuộc thi với mục đích số hóa tương tự như để tài của chúng tôi [61] dành cho chữ Kuzushiji (một loại chữ Nhật cổ) với nhiều lời giải hay cũng như các mô hình mới lạ đến từ nhiều thí sinh, đa phần là họ sẽ sử dụng các kiến trúc quen thuộc trong Object Detection kết hợp với một mạng CNN.

Tuy nhiên, do những nhược điểm của việc triển khai theo mức ký tự đã được đề cập trong những phần trên, nên chúng tôi sẽ hướng tới những cách tiếp cận mà không chỉ giúp các mô hình Recognition có thể nhận dạng được các chữ Hán-Nôm mà còn học được cả ngữ cảnh quanh chúng, thay vì chỉ đơn thuần sử dụng một mạng CNN để nhận dạng riêng lẻ từng ký tự một.

6.3.1. Tiếp cận theo hướng sinh mô tả cho ảnh

Đơn giản và dễ thấy nhất, chúng tôi xem bài toán OCR như một bài toán tạo mô tả bằng ngôn ngữ tự nhiên cho nội dung của hình ảnh (Image Captioning). Trong quá trình hình thành mô tả, ta sẽ liên tục quan sát hình ảnh nhưng đồng thời cũng liên tục tìm cách tạo ra một chuỗi các từ có ý nghĩa. Như vậy, có 2 loại thông tin sẽ cần được xử lý: hình ảnh và mô tả (caption) tương ứng với ảnh đó. Đối với đầu vào là ảnh, chúng tôi sẽ sử dụng một mạng CNN để trích xuất đặc trưng. Còn đối với đầu vào là caption, RNN sẽ được áp dụng để xử lý dữ liệu chuỗi. Câu hỏi đặt ra ở

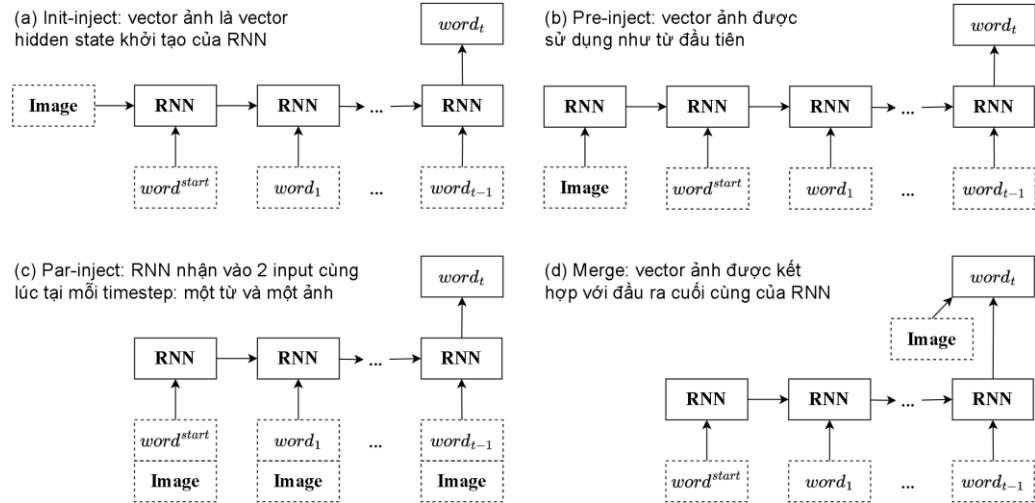
đây là làm sao có thể kết hợp được thông tin giữa 2 nguồn dữ liệu này, tức nên đưa các mẫu thông tin này vào mô hình như thế nào hay theo thứ tự nào?

6.3.1.1. Kiến trúc Injection và Merging

Một nghiên cứu [62] đã trình bày một sự so sánh có hệ thống cho các cách kết hợp khác nhau hay cụ thể hơn là đánh giá việc xử lý chung hay xử lý riêng biệt từng phần thông tin sẽ hiệu quả hơn. Về cơ bản, bài báo sử dụng 2 loại kiến trúc sau:

- **Kiến trúc Chèn (Injection Architecture):** vector đặc trưng của ảnh thu được khi đi qua CNN sẽ được chèn vào RNN, như vậy mỗi step của RNN đều bị ảnh hưởng bởi dữ liệu ảnh. Ngoài ra, việc xử lý ảnh và caption cùng lúc như vậy sẽ cung cấp thông tin thêm cho nhau. Như vậy, thêm câu hỏi khác phát sinh là nên dùng cả ảnh mà bổ trợ cho tất cả RNN step hay nên phân bổ ảnh cho riêng từng step để khớp với độ dài caption đang được sinh ra. Từ đó, kiến trúc Injection lại được chia thành 3 loại nhỏ khác:
 - **Init-inject:** đầu vào cho hidden state đầu tiên của RNN sẽ là vector đặc trưng của ảnh. Đây là một kiến trúc ràng buộc sớm (Early Binding), cho phép RNN sửa đổi biểu diễn ảnh.
 - **Pre-inject:** vector ảnh sẽ được xem như một từ đầu vào, hay đầu vào đầu tiên của RNN sẽ là vector ảnh. Đây cũng là một kiến trúc ràng buộc sớm và cho phép RNN sửa đổi biểu diễn ảnh.
 - **Par-inject:** mỗi step đều là sự tổng hợp của ảnh và từ. Đây là một kiến trúc ràng buộc hỗn hợp (Mixed Binding) và mặc dù cho phép một số sửa đổi trong biểu diễn ảnh nhưng RNN sẽ khó thực hiện điều này, do hidden state của nó sẽ được làm mới liên tục với hình ảnh gốc.
- **Kiến trúc Hợp (Merging Architecture):** xử lý ảnh riêng và xử lý caption riêng, sau đó ghép 2 kết quả này lại, có thể bằng phép cộng (Addition) hay phép ghép (Concatenation). Như vậy RNN sẽ không được tiếp xúc với vector ảnh tại bất kỳ điểm nào. Đây là một kiến trúc ràng buộc muộn (Late Binding), nó sẽ không sửa đổi biểu diễn hình ảnh với mọi step.

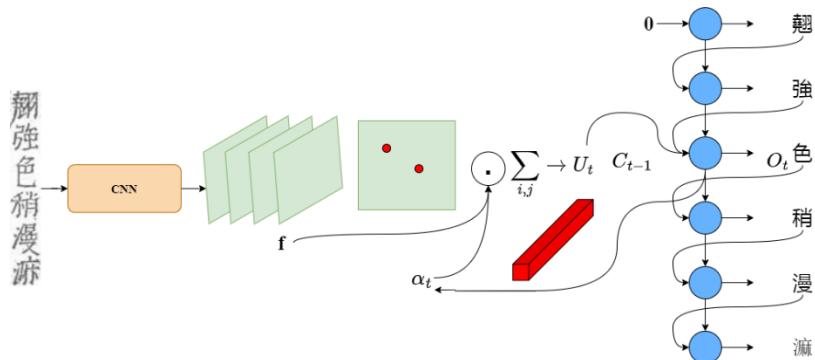
Với các kiến trúc trình bày ở trên, trong bài báo tác giả đã cho thấy kiến trúc Init-inject đạt được nhiều kết quả tốt dựa trên các đánh giá cùng nhiều độ đo khác nhau nên đây sẽ là mô hình đầu tiên chúng tôi lựa chọn cho hướng tiếp cận theo bài toán Image Captioning này. Chúng tôi sẽ gọi mô hình này là Init-inject Captioner từ đây.



Hình 6.5. Các kiến trúc Injection và Merging trong Image Captioning

6.3.1.2. Kiến trúc dựa trên Cơ chế Tập trung

Một cách giải quyết khác của chúng tôi cho việc tiếp cận theo hướng sinh mô tả cho ảnh trong đề tài này là tận dụng sự vượt trội của Cơ chế Tập trung, được lấy cảm hứng từ 2 bài báo [63] và [64]. Bằng cách sử dụng mô hình dựa trên Cơ chế Tập trung (Attention-based) chúng tôi có thể biết được phần nào của ảnh sẽ được mô hình tập trung vào khi nó sinh ra caption. Chúng tôi sẽ xây dựng mô hình này theo kiến trúc Encoder – Decoder và gọi với tên Attention-based Captioner từ đây.

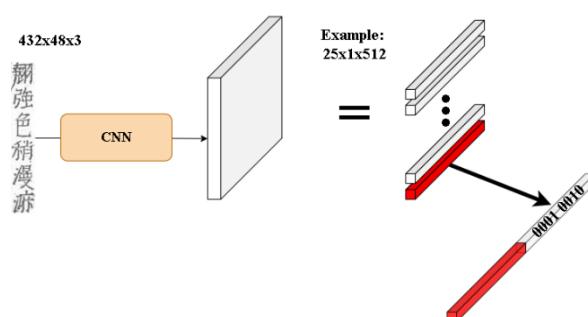


Hình 6.6. Mô hình Image Captioning dựa trên Attention

Nhìn chung, trong luồng đi của mô hình Attention-based Captioner, đầu tiên chúng tôi sử dụng một mạng CNN đóng vai trò là Encoder để trích xuất đặc trưng ảnh. Sau đó vector đặc trưng này cùng với hidden state (được khởi tạo bằng 0) và từ tại ví trí đầu tiên sẽ được truyền vào Decoder, từ đó trả về dự đoán cho từ tiếp theo cùng với hidden state của nó. Hidden state này sau đó sẽ được truyền trở lại vào Decoder cùng với vector ảnh và ký tự ở vị trí tiếp theo. Quy trình sẽ được lặp lại liên tục như vậy cho tới hết câu hay hết chuỗi đầu vào. Ngoài ra, chúng tôi còn sử dụng Teacher Forcing để quyết định đầu vào tiếp theo, đây là kỹ thuật mà trong đó ký tự mục tiêu sẽ được truyền làm đầu vào tiếp theo cho Decoder.

Nhiệm vụ cụ thể của Decoder là sinh ra dự đoán cho ký tự tiếp theo. Ngay khi nhận được vector ảnh hay các feature map từ Encoder và hidden state của ký tự đang xét, Decoder sẽ sử dụng Cơ chế Tập trung để biến đổi chúng thành một vector ngữ cảnh làm đầu vào cho RNN cùng với Embedding của từ hiện tại sau khi đã “tập trung” sự chú ý vào một vùng ảnh nào đó. Như vậy, đầu vào của RNN tại bước t sẽ được tính theo công thức: $x_t = W_c c_{t-1} + W_u u_{t-1}$, trong đó c_{t-1} là embedding của ký tự trước đó và u_{t-1} là vector ngữ cảnh. Phân phối cho dự đoán cuối cùng trên các chữ cái tại thời điểm t sẽ được tính bằng hàm softmax và chữ cái có xác suất cao nhất sẽ là kết quả của dự đoán. Đây còn gọi là Tìm kiếm tham lam (Greedy).

Tuy nhiên, việc sử dụng Cơ chế Tập trung tại đây có thể làm mất tính tuần tự của các pixel. Do đó, chúng tôi sẽ thực hiện nối thêm một vector one-hot của tọa độ vào các feature map, cách làm này được đề xuất trong bài báo [63], qua đó có thể giúp mô hình bổ sung thêm thông tin vị trí của các pixel.

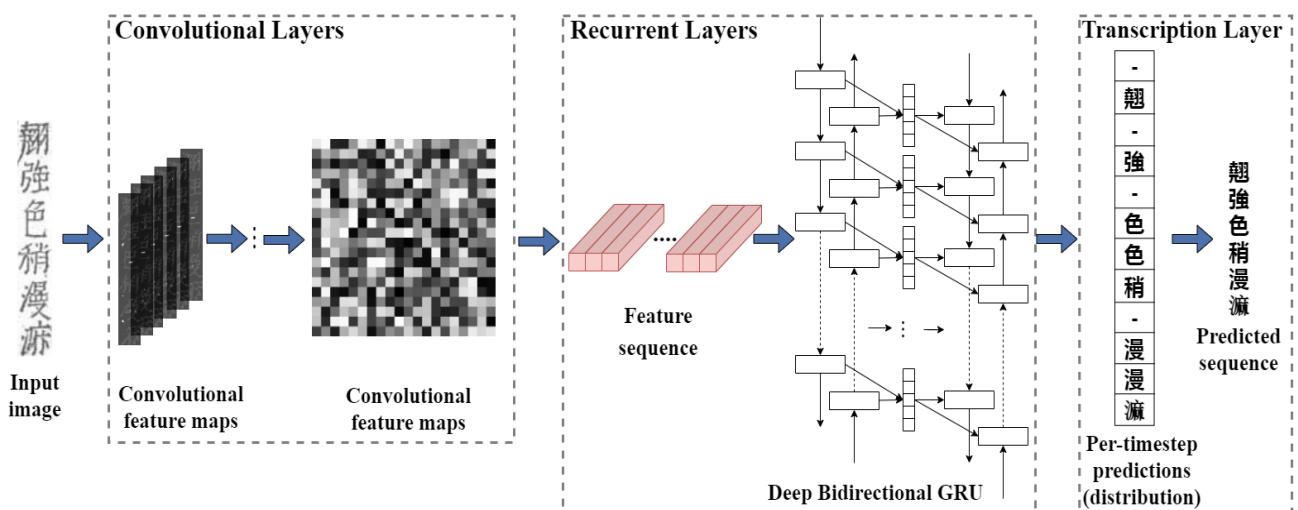


Hình 6.7. Bổ sung tọa độ pixel cho mô hình Attention-based Image Captioning

6.3.2. Mạng Nơ-ron Hồi tiếp Tích chập (CRNN)

Mạng Nơ-ron Hồi tiếp Tích chập (*Convolutional Recurrent Neural Network – CRNN*) [65] được tạo nên dựa trên những cái nhìn cơ bản nhất: bài toán Text Recognition nói một cách đơn giản chính là bài toán nhận dạng chuỗi từ ảnh đầu vào, đối việc xử lý dữ liệu ảnh thì mạng phù hợp nhất thường là CNN, còn đối với vấn đề xử lý trình tự thì phù hợp nhất thường là RNN. Từ đó, CRNN đã ra đời và là sự kết hợp đồng thời giữa 2 mạng CNN và RNN cùng một hàm mất mát CTC. Vì vậy, chúng tôi sẽ gọi mô hình này với tên là CRNNxCTC từ đây.

Mô hình này được thiết kế với mục đích giải quyết các nhiệm vụ nhận dạng trình tự dựa trên hình ảnh, chẳng hạn như các bài toán Scene Text Recognition. Đây cũng là một mô hình phổ biến trong việc nhận dạng chữ in cũng như chữ viết tay và có kết quả rất khả quan dù cho nó có kiến trúc cực kỳ đơn giản.



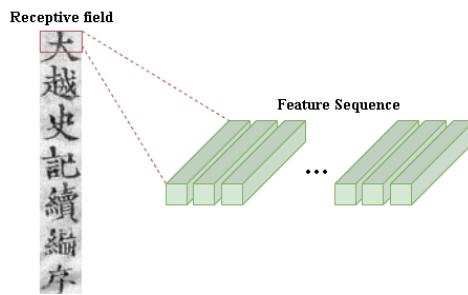
Hình 6.8. Mô hình CRNN (Nguồn [65])

6.3.2.1. Kiến trúc

Từ hình trên có thể thấy kiến trúc của CRNN được chia thành các thành phần chính cũng như có luồng đi theo thứ tự như sau:

- Convolutional Layers: ảnh đầu vào sẽ đi qua một số lớp CNN để trích xuất các đặc trưng, từ đó thu được các feature map. Tiếp theo, các feature map này sẽ được chia thành các cột có chiều rộng 1 pixel hay một chuỗi

các vector đặc trưng (màu đỏ như trong hình bên dưới). Câu hỏi đặt ra là tại sao lại chia các feature map này theo các cột? Câu trả lời liên quan tới khái niệm vùng nhận thức (receptive field), được đề cập trong 4.2.1.1. Đây là vùng trong ảnh đầu vào mà một CNN feature map cụ thể nhìn thấy. Vùng nhận thức của mỗi vector đặc trưng sẽ tương ứng với một vùng hình chữ nhật trong ảnh đầu vào như hình bên dưới, tức ta có thể xem mỗi vector đặc trưng là bộ mô tả của vùng ảnh hình chữ nhật đó.



Hình 6.9. Receptive field trong CRNN (Nguồn [65])

- Recurrent Layers: các vector đặc trưng trên tiếp theo sẽ được truyền vào một mạng RNN 2 chiều sâu (Deep Bidirectional RNN) gồm 2 lớp RNN 2 chiều. Cuối cùng, một hàm softmax được áp dụng để lớp RNN này trả về phân phối xác suất tại mỗi bước thời gian trên tập ký tự. Nhưng các vecto đặc trưng này đôi khi có thể không chứa ký tự hoàn chỉnh, như vecto đặc trưng đầu tiên trong hình 6.9 chỉ chứa một số phần của chữ "大". Do vậy, trong đầu ra của RNN, ta có thể thu được các ký tự lặp lại như có thể thấy trong hình 6.8. Đây còn được gọi là những dự đoán trên mỗi khung hình (per-frame) hay mỗi bước thời gian (per-timestep).
- Transcription Layer: ta chỉ biết kết quả đầu ra cuối cùng (nhãn) chứ không thể biết được dự đoán trên mỗi bước thời gian nên để huấn luyện được mô hình này, ta cần phát triển một cơ chế để chuyển đổi đầu ra mỗi bước thành đầu ra cuối cùng hoặc ngược lại. Mục tiêu của Transcription Layer là lấy chuỗi ký tự lộn xộn trên, trong đó có thể có một số ký tự bị thửa hoặc các ký tự trùng khac, và sử dụng một phương pháp xác suất để thống nhất và làm nó trở nên có ý nghĩa, phương pháp này còn gọi là CTC [66].

6.3.2.2. CTC Loss

Hàm mất mát CTC (*Connectionist Temporal Classification*) [67] được giới thiệu cách khá lâu vào năm 2006 được sử dụng để huấn luyện các mạng sâu nơi mà sự căn chỉnh (alignment) là một vấn đề. Với CTC, ta sẽ không cần phải lo lắng về căn chỉnh hoặc các dự đoán cho mỗi bước thời gian (per-timestep). CTC sẽ xử lý tất cả các căn chỉnh, từ đó ta có thể huấn luyện mô hình chỉ bằng cách sử dụng các đầu ra cuối cùng [66].

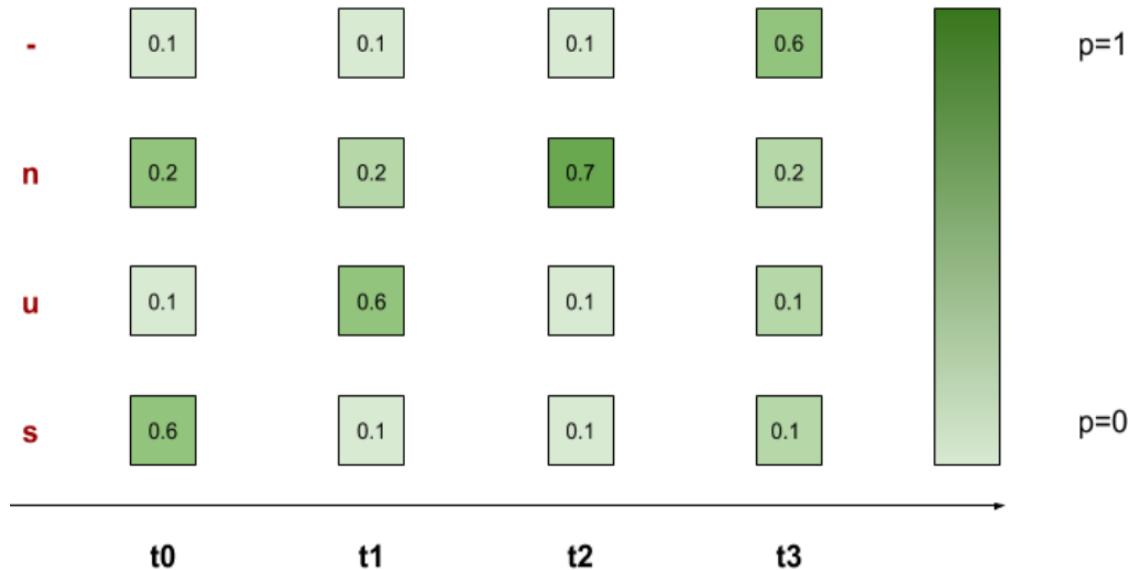
Từ thông tin của nhãn hay văn bản mục tiêu (*Ground Truth Text*) và kết quả đầu ra của RNN, CTC sẽ tính toán mất mát (loss) nhằm mục đích giảm thiểu đường đi bất lợi có khả năng xảy ra cao nhất (negative maximum likelihood). CTC sẽ thử tất cả các căn chỉnh của ground truth và tính score của tổng tất cả các căn chỉnh đó. Căn chỉnh của ground truth sẽ được phát sinh bằng cách thêm một ký tự rỗng “-” (blank token) và lặp lại bất kỳ ký tự nào có trong ground truth.

Mô hình sẽ học để dự đoán những căn chỉnh trên, sau đó ta phải thực hiện decode để đưa ra chuỗi dự đoán cuối cùng bằng cách gộp những ký tự lặp lại liên tiếp nhau thành một ký tự, sau đó xóa hết tất cả ký tự rỗng. Như vậy, để ánh xạ chuỗi đầu vào $X = [x_1, x_2, \dots, x_T]$ sang chuỗi đầu ra tương ứng $Y = [y_1, y_2, \dots, y_U]$, CTC sẽ tính tổng tất cả các xác suất có thể xảy ra khi căn chỉnh của text có trong ánh bằng cách sử dụng công thức (6.11) [68].

$$p(Y|X) = \sum_{A \in A_{X,Y}} \prod_{t=1}^T p_t(a_t|X) \quad (6.11)$$

Xác suất có điều kiện trong CTC	Tổng tất cả các căn chỉnh đúng	Xác suất cho một căn chỉnh theo từng timestep
---------------------------------	--------------------------------	---

Ví dụ với ground truth là từ “sun” và RNN sẽ dự đoán cho 4 bước thời gian thì những căn chỉnh có thể có của ground truth sẽ là: -sun, s-un, su-n, sun-, suun, ssun, sunn, với các xác suất ví dụ cho mỗi bước thời gian như hình bên dưới. Bất kỳ căn chỉnh nào được dự đoán đều là một dự đoán đúng. Vì vậy, hàm loss cần được tối ưu chính là tổng các căn chỉnh [69].



Hình 6.10. Xác suất theo các bước thời gian của từ “sun” để tính toán CTC

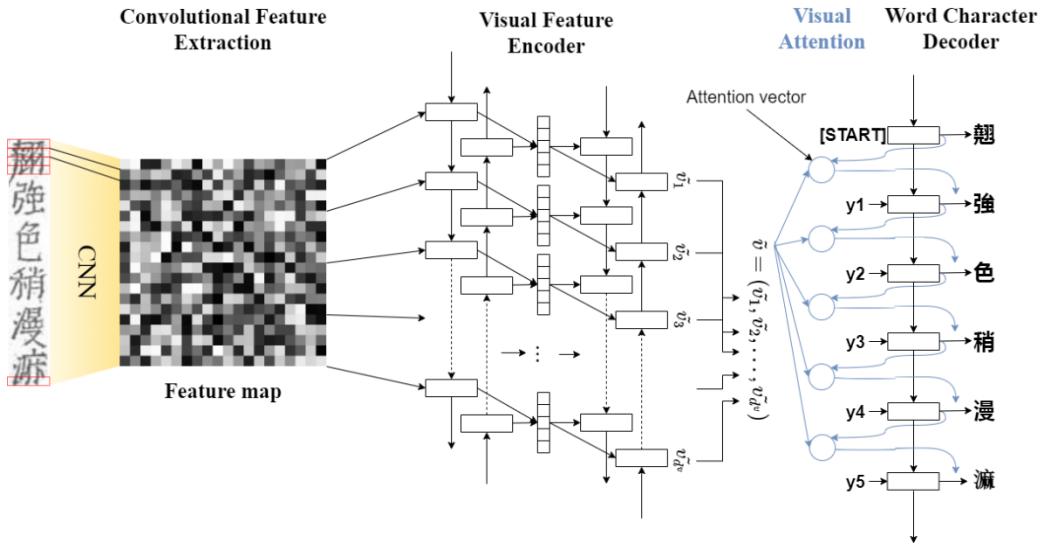
Với từ “sun”, tổng xác suất của 7 căn chỉnh theo hình trên sẽ là: $p('sun')$

$$\begin{aligned}
 &= p('sun') + p('s-un') + p('su-n') + p('sun-') + p('ssun') + p('suun') + p('sunn') \\
 &= 0.1*0.1*0.1*0.2 + 0.6*0.1*0.1*0.2 + 0.6*0.6*0.1*0.2 + 0.6*0.6*0.7*0.6 + \\
 &\quad 0.6*0.1*0.1*0.2 + 0.6*0.6*0.1*0.2 + 0.6*0.6*0.7*0.2 = 0.2186
 \end{aligned}$$

6.3.3. Tiếp cận theo hướng Seq2Seq trong dịch máy

Kiến trúc CRNN sử dụng CTC Loss có một hạn chế là ta phải cẩn thận điều chỉnh kiến trúc mô hình để kích thước của vùng nhận thức khớp với số lượng ký tự tối đa có thể dự đoán. Một bổ sung phổ biến cho mô hình CRNNxCTC có thể được sử dụng để cải thiện dự đoán của văn bản trong ảnh đầu vào là một Cơ chế Tập trung. Trong hướng tiếp cận này, như thường lệ, chúng tôi đầu tiên sẽ sử dụng các mạng CNN để trích xuất đặc trưng ảnh. Sau đó, các đặc trưng này sẽ được chuyên thành chuỗi và truyền qua mạng RNN để có được kết quả cho Cơ chế Tập trung xử lý.

Mô hình chúng tôi sử dụng trong quá trình này được lấy cảm hứng và có cách hoạt động tương tự mô hình Attention Seq2Seq cho bài toán dịch máy [52]. Với một bài toán dịch máy từ tiếng Việt sang Anh, ta cần mã hóa một chuỗi tiếng Việt thành một vector đặc trưng, còn trong mô hình này, dữ liệu đầu vào sẽ là một ảnh. Chúng tôi sẽ gọi mô hình này với tên là CNNxSeq2Seq từ đây.



Hình 6.11. Mô hình CNN kết hợp Attention-Seq2Seq

Decoder sẽ sử dụng một Attention head để “tập trung” có chọn lọc vào các phần của chuỗi đầu vào. Lớp Tập trung này tương tự như một lớp gộp trung bình (Average Pooling) nhưng nó biểu diễn một trung bình có trọng số (weighted average). Nhiệm vụ của Decoder lúc này là sinh ra các dự đoán cho ký tự tiếp theo bằng cách nhận vào đầu ra hoàn chỉnh của Encoder, sau đó nó sẽ sử dụng RNN để theo dõi những gì nó đã tạo ra cho tới bước hiện tại. Đầu ra của RNN sẽ được dùng làm query để thực hiện “tập trung” trên đầu ra của Encoder, từ đó hình thành nên một vectơ ngữ cảnh. Tiếp theo Decoder sẽ kết hợp đầu ra của RNN và vectơ ngữ cảnh này để sinh ra vector tập trung (attention vector), kết quả dự đoán cho ký tự tiếp theo sẽ dựa trên vector này. Công thức cho quy trình trên của mô hình trong hướng tiếp cận này sẽ lần lượt như sau [70]:

- $score(h_t, h_s) = v_a^t \tanh(W_1 h_t + W_2 h_s)$ [Bahdanau's additive attention] (6.12)

- $a_{ts} = softmax(score(h_t, h_s))$ [Attention weights] (6.13)

- $c_t = \sum_s a_{ts} h_s$ [Context vector] (6.14)

- $a_t = f(c_t, h_t) = \tanh(W_c [c_t; h_t])$ [Attention vector] (6.15)

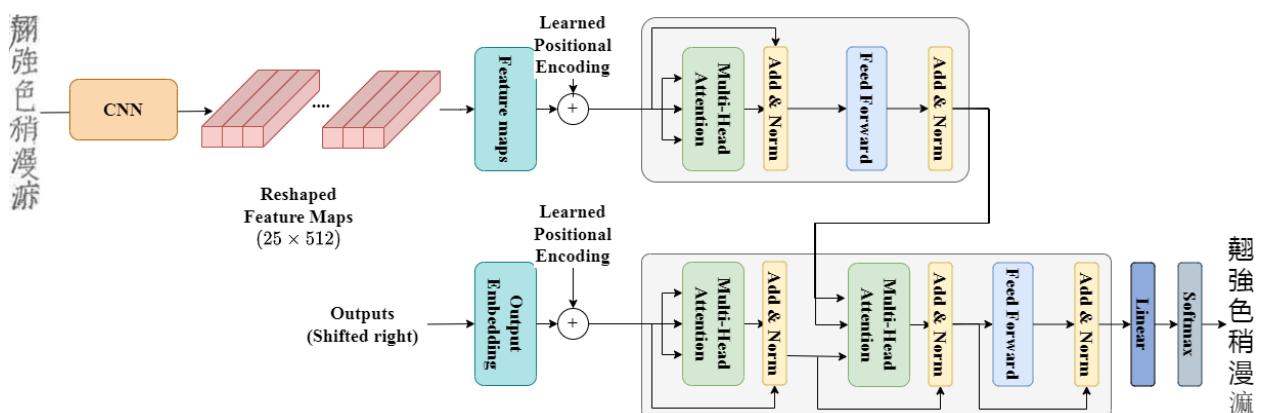
Việc huấn luyện mô hình này tương tự như huấn luyện mô hình Seq2Seq, chúng sử dụng Cross-entropy để tối ưu thay vì CTC Loss như CRNN. Nghĩa là tại mỗi bước thời gian, mô hình sẽ dự đoán một ký tự để tính loss so với nhãn và cập nhật

lại trọng số của mô hình. Tuy nhiên đối với mô hình này, các câu ngắn sẽ thường hoạt động tốt hơn, nhưng nếu đầu vào quá dài, mô hình sẽ mất tập trung theo đúng nghĩa đen và ngừng cung cấp các dự đoán hợp lý. Có 2 lý do chính cho việc này:

- Mô hình được huấn luyện sử dụng Teacher Forcing sẽ cung cấp ký tự đúng tại mỗi bước, bất kể kết quả dự đoán là gì. Do đó, mô hình có thể trở nên mạnh mẽ hơn nếu được cung cấp thêm các dự đoán của chính nó.
- Mô hình chỉ có thể biết được đầu ra trước đó của nó thông qua trạng thái của RNN. Nếu trạng thái này bị hỏng, sẽ không có cách nào để mô hình phục hồi. Transformer đã giải quyết điều này bằng cách sử dụng khả năng tự chú ý (Self-Attention) trong Encoder và Decoder.

6.3.4. Các mô hình TransformerOCR

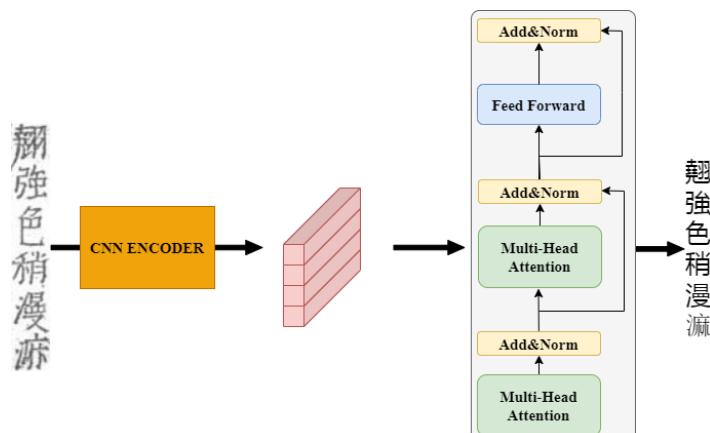
Kiến trúc Seq2Seq trên vẫn còn tồn tại một số nhược điểm nhất định khác như đã được đề cập trong 4.2.3.5. Vì vậy, một cách phổ biến khác để tăng độ chính xác cho các kiến trúc của bài toán Recognition là tận dụng Transformer. Ngoài việc khắc phục được những nhược điểm của mô hình Seq2Seq, Transformer cũng đi kèm với rất nhiều ưu điểm khác như đã được đề cập trong 4.2.4.5. Kiến trúc này sẽ có chức năng tương tự như các mạng RNN nhưng điểm khác biệt là nó không yêu cầu xử lý dữ liệu đầu vào theo thứ tự (nghĩa là từ đầu đến cuối). Điều này đồng thời cũng có thể làm giảm đáng kể thời gian cần thiết để huấn luyện một kiến trúc Recognition như vậy. Chúng tôi sẽ gọi mô hình này với tên là CNNxTransformer từ đây.



Hình 6.12. Mô hình CNN kết hợp kiến trúc Transformer

Ở mô hình trên, chúng tôi sẽ chia nhỏ vấn đề thành 2 module con: một module trích xuất đặc trưng và một module Transformer. Feature map sẽ được xem như một Embedding và làm đầu vào cho Encoder để tạo ra một biểu diễn mới cho đầu vào. Decoder sẽ sử dụng đầu ra của Encoder và văn bản mục tiêu làm đầu vào để có gắng học cách sinh đầu ra đúng. Khi mô hình dự đoán từng vị trí trong chuỗi đầu vào, Self-Attention cho phép xem xét các vị trí trước đó để đưa ra dự đoán cho vị trí tiếp theo tốt hơn.

Bên cạnh mô hình sử dụng CNN kết hợp toàn bộ kiến trúc Transformer trên, chúng tôi cũng xây dựng thêm một mô hình khác dựa trên kiến trúc Transformer mà không cần sử dụng các biểu diễn trình tự trung gian. Chúng tôi thực hiện kết nối trực tiếp các đặc trưng từ CNN với Transformer Decoder đóng vai trò như một bộ giải mã trình tự dựa trên Cơ chế Tập trung. Chúng tôi sẽ gọi mô hình này là CNNxTransformerDecoder từ đây.



Hình 6.13. Mô hình CNN kết hợp chỉ Transformer Decoder

Transformer là một mô hình tự động hồi quy (auto regressive) như được đề cập trong 4.2.4.5, nghĩa là nó sẽ đưa ra dự đoán từng phần một và sử dụng kết quả đầu ra từ bước đầu cho đến bước hiện tại để quyết định phải làm gì tiếp theo. Do vậy, văn bản mục tiêu cần được chia làm 2 phần: một phần được giữ nguyên dùng làm đầu vào cho Decoder, phần còn lại sẽ được dịch (shift) sang 1 đơn vị để làm mục tiêu học. Ngoài ra, chúng tôi cũng sử dụng Teacher Forcing để truyền đầu ra đúng cho bước thời gian tiếp theo, bất kể mô hình dự đoán được gì ở bước hiện tại [53].

Chương 7. CÀI ĐẶT THỬ NGHIỆM

7.1. Triển khai cho bài toán Text Detection

Với bài toán này, chúng tôi sẽ thử nghiệm 2 mô hình EAST và DBNet, đại diện cho 2 phương pháp Regression-based và Segmentation-based cho các Page của bộ dữ liệu NomNaOCR. Chúng tôi sẽ sử dụng Resnet 18 làm backbone hay mô hình cơ sở dùng để trích xuất đặc trưng ảnh cho cả 2 mô hình trên, cùng batch size được thiết lập là 8, và thuật toán tối ưu (optimizer) là Adam. Các thử nghiệm cho bài toán Text Detection này sẽ được thực hiện trên môi trường của Google Colab Pro sử dụng với cấu hình 25GB RAM và một GPU NVIDIA T4.

Đối với EAST, chúng tôi sẽ huấn luyện mô hình này trong 115 epoch, sử dụng learning rate là 0.001 và được warm-up sau 10 epoch với ảnh đầu vào sẽ được resize về kích thước 512x512. Cuối cùng ở bước hậu xử lý (Post-process) để hình thành kết quả cuối cùng từ đầu ra mô hình, các hình học (geometries) được giữ lại sau khi thỏa mãn một ngưỡng nhất định (threshold) là 0.2, sẽ được hợp nhất bằng thuật toán NMS.

Đối với DBNet, chúng tôi sẽ huấn luyện mô hình này trong 15 epoch, sử dụng learning rate được định thời theo hàm cosine với giá trị khởi tạo là 0.001 và được warm-up sau 2 epoch. Tất cả ảnh đầu vào sẽ được đưa về một kích thước chung là 960x960 và được thực hiện tăng cường dữ liệu với các thao tác bao gồm: quay ngẫu nhiên với biên độ góc trong khoảng (-10, 10) và resize kích thước ngẫu nhiên. Cuối cùng, với bước Post-process cho DBNet, chúng tôi sẽ thực hiện các cài đặt mà mô hình này yêu cầu để có thể thu được các bounding box một cách hiệu quả:

- Thresh = 0.3: ngưỡng yêu cầu cho kết quả nhị phân hóa (binarization) của segmentation map.
- Box thresh = 0.5: ngưỡng để lọc các box đầu ra (dưới ngưỡng này sẽ bị loại).
- Max candidates = 1000: số lượng tối đa các bounding box đầu ra.
- Unclip ratio = 2.5: tỷ lệ mở rộng của bounding box.

7.2. Triển khai cho bài toán Text Recognition

Chúng tôi sẽ tiến hành huấn luyện các mô hình với kích thước batch là 32 và áp dụng Early Stopping để dừng quá trình huấn luyện lại nếu độ lỗi (loss) của tập Validate không được cải thiện ít nhất 0.001 đơn vị sau 5 epoch. Ngoài CRNN sử dụng hàm mục tiêu là CTC, các mô hình còn lại đều sử dụng Cross-entropy để tính loss và cập nhật lại trọng số. Các thử nghiệm sẽ được thực hiện trên hệ thống phần cứng với cấu hình 64GB RAM cùng một GPU NVIDIA RTX 2080 SUPER 8GB.

7.2.1. Các giai đoạn huấn luyện

Chúng tôi chia việc huấn luyện các mô hình Recognition làm 3 giai đoạn (phase):

- Tiền huấn luyện (Pre-training): trong khi xây dựng bộ dữ liệu NomNaOCR, chúng tôi cũng đồng thời xây dựng các mô hình Text Recognition và sử dụng bộ dữ liệu Synthetic Nom String để huấn luyện, cũng như để tìm ra các cài đặt tối ưu cho các mô hình trên trong lúc bộ dữ liệu chính vẫn đang được hoàn thiện. Giai đoạn này cũng có thể được xem là một cách tăng cường ảnh.
- Tinh chỉnh (Fine-tuning): sau khi NomNaOCR được hoàn thành, chúng tôi sẽ cho mô hình được huấn luyện tiếp trên bộ dữ liệu thật này từ các trọng số (weights) đã học được trong giai đoạn Pre-training.
- Huấn luyện lại (Retraining): cho mô hình được huấn luyện từ đầu trên dữ liệu thật và so sánh kết quả với khi Fine-tuning.

7.2.2. CNN backbone

Nhìn chung, các mô hình chúng tôi triển khai đều bao gồm 2 phần chính: một phần backbone để trích xuất đặc trưng ảnh và một phần để xử lý ngôn ngữ. Trong đề tài này, chúng tôi sẽ xây dựng một mạng CNN đơn giản để thực hiện việc trích xuất đặc trưng trên, với kích thước chiều cao của vùng nhận thức được điều chỉnh cho phù hợp với chiều dài văn bản tối đa trong bộ dữ liệu NomNaOCR. Từ đó, chúng tôi sẽ mô hình hóa feature map như một chuỗi các đặc trưng và dùng làm đầu vào cho phần xử lý ngôn ngữ.

Ngoài ra, tác giả của IHR-NomDB [9] cũng đã chỉ ra rằng việc sử dụng ảnh RGB với độ phân giải cao sẽ giúp mô hình Deep Learning dành cho ảnh viết tay có hiệu suất tốt hơn. Như vậy, các ảnh đầu vào hay các Patch sẽ được xử lý dưới dạng 3 kênh màu RGB và được resize về một kích thước chung nhưng tỷ lệ khung hình vẫn được đảm bảo cùng nội dung của Patch không bị ảnh hưởng.

Cụ thể, chúng tôi sẽ sử dụng kích thước được đề xuất trong IHR-NomDB là 432x48x3, từ đó có được đầu vào cho lớp Convolution đầu tiên của mạng CNN. Bên cạnh đó, chúng tôi cũng xếp chồng một loạt các khối (Block) gồm các bộ ConvBnRelu với 3 lớp Convolution, Batch Normalization và ReLU, cùng các lớp Pooling tương ứng lần lượt vào mạng CNN trên để đảm bảo đầu ra có chiều rộng (width) là 1. Cuối cùng, đầu ra của CNN sẽ được định hình lại (Reshape) như một chuỗi các đặc trưng hay feature map, để khớp với đầu vào cho phần xử lý ngôn ngữ của mô hình đang sử dụng.

Bảng 7.1. Cài đặt CNN backbone cho Text Recognition

Lớp (Layers)		Cấu hình (Configurations)	Kích thước output
Đầu vào (Input)		Ảnh RGB kích thước 432x48	(432, 48, 3)
Block 1	ConvBnRelu	64 filters, 3x3 kernel, có padding	(432, 48, 64)
	MaxPooling	2x2 kernel	(216, 24, 64)
Block 2	ConvBnRelu	128 filters, 3x3 kernel, có padding	(216, 24, 128)
	MaxPooling	2x2 kernel	(108, 12, 128)
Block 3	ConvBnRelu	256 filters, 3x3 kernel, có padding	(108, 12, 256)
	ConvBnRelu	256 filters, 3x3 kernel, có padding	(108, 12, 256)
	MaxPooling	2x2 kernel	(54, 6, 256)
Block 4	ConvBnRelu	512 filters, 3x3 kernel, có padding	(54, 6, 512)
	ConvBnRelu	512 filters, 3x3 kernel, có padding	(54, 6, 512)
	MaxPooling	2x2 kernel	(27, 3, 512)
Block 5	ConvBnRelu	512 filters, 2x2 kernel, 0 padding	(26, 2, 512)
	ConvBnRelu	512 filters, 2x2 kernel, 0 padding	(25, 1, 512)

7.2.3. Cài đặt phần Xử lý ngôn ngữ

Các đầu vào văn bản sẽ được tiền xử lý bằng cách chỉ giữ lại các chuỗi nào chứa các ký tự Hán-Nôm (không chứa các chữ cái Latin, số, các dấu câu, dấu ngoặc hay các ký tự không cần thiết). Nhìn chung các mô hình Machine Learning hay Deep Learning thường sẽ không xử lý trực tiếp với dạng văn bản nên chúng tôi sẽ quy đổi (encode) các ký tự trong các chuỗi trên về dạng số (token).

Ngoại trừ CRNN sử dụng feature map làm đầu vào trực tiếp cho phần xử lý ngôn ngữ, các mô hình còn lại sẽ thêm 2 token “[START]” và “[END]” để biểu diễn sự bắt đầu và kết thúc của văn bản, cuối cùng, các token sẽ tiếp tục được encode sang dạng vector bằng một lớp Embedding với chiều đầu ra là 512. Ngoài ra, các chuỗi đầu vào sẽ được đệm đến chiều dài tối đa trong bộ dữ liệu bằng một token “[PAD]” để tất cả đều có chung một độ dài cố định.

Ngoài các mô hình tận dụng kiến trúc Transformer cho phần xử lý ngôn ngữ, đối với các mô hình sử dụng RNN, chúng tôi sẽ sử dụng GRU để thử nghiệm vì đơn giản đây là một RNN mạnh mẽ với chỉ một vector trạng thái ẩn thay vì 2 vector trạng thái như LSTM (trạng thái ẩn và trạng thái ô). Điều này sẽ khiến cho việc cài đặt các kiến trúc trở nên phức tạp hơn cũng như tổng tham số và thời gian cần để hội tụ có thể cao hơn.

7.2.4. Thuật toán tối ưu (Optimizer)

Chúng tôi sẽ thử nghiệm với 2 thuật toán tối ưu là Adam và Adadelta. Với Adam, đây là một optimizer phỏ biến và thường có hiệu quả tốt khi huấn luyện các mô hình Deep Learning. Chúng tôi sẽ khởi tạo một giá trị learning rate là 0.0002 khi sử dụng Adam cùng một chiến lược giảm learning rate đi một nửa mỗi khi độ lỗi của tập Validate không được cải thiện sau 2 epoch.

Còn với Adadelta [71], đây là một phản hồi rộng mạnh mẽ hơn của Adagrad giúp điều chỉnh learning rate. Nhờ đó, Adadelta có thể tiếp tục học ngay cả khi nhiều cập nhật đã được thực hiện. Adadelta có xu hướng được hưởng lợi ích từ các giá trị learning rate ban đầu cao hơn so với các thuật toán tối ưu hóa khác. Ở đây,

chúng tôi sẽ khởi tạo giá trị ban đầu này chính xác với trong bài báo gốc là 1.0. Như vậy, khi sử dụng Adadelta, chúng tôi sẽ không cần quan tâm tới việc điều chỉnh giá trị learning rate cho phù hợp với mô hình cũng như quá trình huấn luyện.

7.2.5. Các thông số khác

Ngoài các cài đặt chung được trình bày trong những phần trên, các mô hình dựa trên RNN vẫn còn một siêu tham số (hyperparameter) khác cũng không kém phần quan trọng và cần được điều chỉnh riêng để đạt được hiệu suất tốt cho quá trình huấn luyện chính là số nơ-ron (units) của GRU, đóng vai trò then chốt để quyết định các token cho đầu ra.

Bảng 7.2. Cài đặt GRU units cho các mô hình Recognition

Mô hình	GRU units
Init-inject Captioner	512
Attention-based Captioner	1024
CRNNxCTC	256
CNNxSeq2Seq	256

Đối với các mô hình Recognition dựa trên kiến trúc Transformer, chúng tôi sẽ thực hiện điều chỉnh một vài hyperparameter được đề cập trong bài báo gốc “Attention is all you need” của kiến trúc này [51], như sau:

Bảng 7.3. Cài đặt cho các mô hình Recognition tận dụng kiến trúc Transformer

Mô hình TransformerOCR	N	d_{model}	d_{ff}	h	P_{drop}
CNNxTransformer	2	512	512	1	0.1
CNNxTransformerDecoder	2	512	512	1	0.1

Trong đó:

- N : số layers cho các lớp Encoder hoặc Decoder.
- d_{model} : tương tự chiều đầu ra của lớp Embedding và cũng đồng thời là số lượng nơ-ron cho các lớp fully connected được sử dụng.

- d_{ff} : số nơ-ron cho riêng lớp fully connected đầu tiên của Position-wise Feed-Forward Network.
- h : số đầu (heads) sẽ sử dụng cho các khối Multi-Head Attention, hay nói cách khác là số lượng cơ chế Attention sẽ được chạy song song.
- P_{drop} : tỷ lệ dropout sẽ sử dụng

7.2.6. Thử nghiệm với các Kết nối tắt

Trong quá trình thực hiện giai đoạn Pre-training, chúng tôi nhận thấy khi cho feature map kết nối tắt với một lớp X có cùng các kích thước và nằm xa nó nhất có thể, sẽ cải thiện đáng kể hiệu suất mô hình. Vì vậy, chúng tôi cũng sẽ thử nghiệm thêm 2 phương pháp Kết nối tắt cơ bản, được đề cập trong [4.2.1.3](#) là Addition và Concatenation cho các mô hình khả thi nhất (tồn tại lớp X nói trên) bao gồm: CRNNxCTC cùng CNNxSeq2Seq với lớp X là lớp GRU 2 chiều thứ 2 của mô hình và CNNxTransformer với lớp X là lớp Transformer Encoder cuối cùng.

Chương 8. ĐÁNH GIÁ VÀ KẾT QUẢ

8.1. Phương pháp đánh giá

8.1.1. Metrics đánh giá Text Detection và End-to-End

Để đánh giá kết hợp 2 tác vụ phát hiện và nhận dạng các ký tự trong ảnh, phương pháp thường được sử dụng là Giao điểm qua liên kết (*Intersection Over Union - IoU*) kết hợp với Các từ được công nhận chính xác (*Correctly Recognized Words – CRW*), tức phương pháp này sẽ đánh giá dựa trên hai chỉ số là IoU và CRW.

Với giai đoạn phát hiện văn bản, phương pháp IoU chỉ chấp nhận bounding box được dự đoán là đúng khi và chỉ khi giá trị IoU cho 2 bounding box được dự đoán thỏa một ngưỡng nhất định (thường > 0.5). Mặc dù IoU được sử dụng rộng rãi nhưng cách tính của nó lại không phù hợp đối với các bài toán cần độ chi tiết và chính xác cao, điều này là rất quan trọng đối với các tác vụ OCR.

Với giai đoạn nhận dạng văn bản trong ảnh, phương pháp CRW được sử dụng như một cách đánh giá nhị phân đơn thuần, khi tất cả các văn bản được nhận dạng trùng hoàn toàn với nhau thì sẽ được tính là 1, ngược lại sẽ là 0. Vì vậy, giới hạn của phương pháp này là không thể đưa ra các chỉ số đánh giá khác nhau cho một kết quả nhận dạng vô lý và một kết quả gần như chính xác.

Từ trên có thể thấy, phương pháp đánh giá IoU kết hợp với CRW có nhiều mặt hạn chế. Vì vậy ở đây, chúng tôi quyết định sẽ sử dụng một phương pháp mới có tên là CLEval [57]. Phương pháp này sẽ giúp đánh giá chính xác hơn cho cả 2 giai đoạn đã đề cập. Cụ thể, nó có thể đánh giá chính xác trong trường hợp 2 bounding box khi kết hợp lại chính là nhau cần phát hiện. Trong khi đó phương pháp IoU lại có thể không chấp nhận một trong 2 hoặc cả 2 box vì ngưỡng được chọn thường lớn hơn 0.5. Còn ở giai đoạn nhận dạng văn bản, phương pháp sẽ trả về các chỉ số khác nhau tùy thuộc vào độ tương đồng của dự đoán so với nhau.

Ngoài ra, phương pháp này còn có thể đánh giá được cho riêng bài toán phát hiện văn bản, vì đặc thù của bài toán này là không có các nhãn văn bản trong các

bounding box nêu tùy vào bài toán mà CLEval sẽ có sự khác nhau trong các thành phần tính toán của nó. Phương pháp đánh giá CLEval được tính toán theo các công thức bên dưới:

$$Recall = \frac{\sum_{i=1}^{|G|} (CorrectNum_i^G - GranulPenalty_i^G)}{\sum_{i=1}^{|G|} TotalNum_i^G} \quad (8.1)$$

$$Precision = \frac{\sum_{j=1}^{|D|} (CorrectNum_j^D - GranulPenalty_j^D)}{\sum_{j=1}^{|D|} TotalNum_j^D} \quad (8.2)$$

$$F1 = 2 \times \frac{Recall \times Precision}{Recall + Precision} \quad (8.3)$$

Trong đó:

- G : tập hợp các box và các ký tự trong dữ liệu, với i là kích thước của tập.
- D : tập hợp các box và các ký tự dự đoán được, với j là kích thước của tập.

Đối với bài toán End-to-End:

- $CorrectNum_i^G$: số lượng ký tự có trong nhãn G_i nằm trong D_j .
- $CorrectNum_j^D$: số lượng ký tự có trong dự đoán D_j nằm trong nhãn G_i .
- $GranulPenalty_i^G$: chỉ số phạt thể hiện mức độ làm mất thông tin kết nối của kết quả, tương ứng với số lượng bounding box trong D khớp với G_i .
- $GranulPenalty_j^D$: chỉ số phạt thể hiện mức độ làm mất thông tin kết nối của kết quả, tương ứng với số lượng bounding box trong G khớp với D_j .
- $TotalNum_i^G$: độ dài văn bản trong nhãn G_i .
- $TotalNum_j^D$: độ dài văn bản trong nhãn D_j .

Đối với riêng bài toán Text Detection:

- $CorrectNum_i^G$: số lượng các điểm PCC của G_i nằm trong D_j .
- $CorrectNum_j^D$: số lượng các điểm PCC tích lũy của D_j nằm trong kết quả khớp với nhãn G_i .

- $GranulPenalty_i^G$: chỉ số phạt thẻ hiện mức độ làm mất thông tin kết nối của kết quả, tương ứng với số lượng bounding box trong D khớp với G_i .
- $GranulPenalty_j^D$: chỉ số phạt thẻ hiện mức độ làm mất thông tin kết nối của kết quả, tương ứng với số lượng bounding box trong G khớp với D_j .
- $TotalNum_i^G$: độ dài văn bản trong nhãn G_i .
- $TotalNum_j^D$: độ dài ước tính cho số ký tự trong G_i khớp với D_j .

8.1.2. Metrics đánh giá với riêng Text Recognition

Chúng tôi sử dụng phương pháp đánh giá tương tự với các công trình liên quan trước đó cho bài toán Recognition theo mức chuỗi để có cái nhìn khách quan về mô hình cũng như về bộ dữ liệu. Các phương pháp đánh giá sẽ được dùng bao gồm: Độ chính xác mức ký tự (*Character Accuracy* – char_acc), Độ chính xác mức chuỗi (*Sequence Accuracy* – seq_acc), và Tỷ lệ lỗi ký tự (*Character Error Rate* - CER).

Đối với phương pháp đánh giá Character Accuracy, mỗi ký tự dự đoán được sẽ được so khớp với mỗi ký tự đúng tương ứng trong Patch đầu vào:

$$Accuracy_{character} = \frac{\sum_i \sum_j TP_{i,j}(character)}{\sum_i Total_i(character)} \quad (8.4)$$

Trong đó:

- $TP_{i,j}(character)$: số ký tự dự đoán khớp với ký tự thật trong chuỗi được gán nhãn. Với i là chuỗi thứ i của dữ liệu, j là ký tự thứ j của chuỗi dự đoán.
- $Total_i(character)$: tổng số ký tự trong một chuỗi được gán nhãn.

Đối với phương pháp đánh giá Sequence Accuracy, chuỗi được dự đoán phải khớp hoàn toàn với chuỗi được gán nhãn:

$$Accuracy_{sequence} = \frac{\sum_i TP_i(sequence)}{Total sequence} \quad (8.5)$$

Trong đó:

- $TP_i(sequence)$: số chuỗi được dự đoán khớp với chuỗi được gán nhãn.
- $Total sequence$: tổng số chuỗi trong bộ dữ liệu.

Cuối cùng là phương pháp đánh giá Character Error Rate, phương pháp này dựa trên cách tính khoảng cách Levenshtein giữa 2 từ (Levenshtein Distance hay Edit Distance) [72]. Theo đó, ta cần đếm số lượng các hành động (thêm, xóa hoặc thay thế) tối thiểu cần để biến đổi một chuỗi ký tự đầu vào thành chuỗi ký tự đầu ra của mô hình OCR.

$$CER = \frac{S + D + I}{N} \quad (8.6)$$

Trong đó:

- S : số lần thay thế.
- D : số lần xóa.
- I : số lần chèn.
- N : số lượng các ký tự trong văn bản gốc.

8.2. Kết quả thử nghiệm

8.2.1. Kết quả bài toán Text Detection

Bộ dữ liệu NomNaOCR được xây dựng trên các tác phẩm là thơ và văn xuôi. Đối với thơ, cấu trúc của các tác phẩm này sẽ khá tương đồng nhau giữa các Page do chúng có cùng thể thơ lục bát. Trong khi đó, các tác phẩm văn xuôi thường có cấu trúc không đồng nhất và phức tạp hơn nhiều, như được trình bày trong [Chương 5](#). Bên cạnh đó, Text Detection có thể hiểu đơn giản là một bài toán quan tâm vào vị trí của các văn bản xuất hiện trong ảnh, vì vậy chúng tôi sẽ tiến hành đánh giá bài toán này bằng phương pháp cũng như các metric được đề cập trong [8.1.1](#) theo 3 hướng: trên toàn bộ ảnh, trên ảnh của các tác phẩm thơ, và trên các ảnh văn xuôi của tập Validate.

Ngoài ra, chúng tôi cũng nhận thấy giữa từng tác phẩm trong bộ dữ liệu NomNaOCR gồm Lục Vân Tiên, Truyện Kiều (các bản năm 1866, 1871, và 1872), và 5 bản của Đại Việt Sử Ký Toàn Thư có sự khác nhau về điều kiện ảnh hoặc cấu trúc văn bản xuất hiện trong ảnh, nên chúng tôi cũng sẽ tiến hành phân tích chi tiết cho từng tác phẩm.

8.2.1.1. Kết quả tổng quan

Bảng 8.1. Kết quả tổng quan cho Text Detection

	Mô hình	Precision	Recall	F1-score
Toàn bộ	DBNet	0.9991	0.9939	0.9965
	EAST	0.9910	0.9844	0.9877
Tho	DBNet	1.0000	0.9998	0.9999
	EAST	0.9994	0.9959	0.9976
Văn xuôi	DBNet	0.9989	0.9926	0.9957
	EAST	0.9891	0.9818	0.9855

Bảng 8.1 thể hiện kết quả tổng quan dựa trên các độ đo Precision, Recall, và F1-score cho 3 hướng phân tích đã đề cập (Toàn bộ, Tho, và Văn xuôi). Các mô hình phát hiện văn bản chúng tôi sử dụng là DBNet và EAST cho kết quả rất tốt với hiệu suất các độ đo đều trên 98%. Trong đó, mô hình DBNet cho kết quả tốt hơn so với EAST khi vượt trội trên cả 3 hướng đánh giá với F1-score lần lượt là 0.9965 trên toàn bộ ảnh, 0.999 trên các ảnh tho, và 0.9957 trên các ảnh văn xuôi cho tập Validate của bộ dữ liệu NomNaOCR. Ngoài ra, cũng có thể thấy rằng cả 2 mô hình DBNet và EAST đều dự đoán các ảnh tho tốt hơn so với văn xuôi. Vì vậy chúng tôi kết luận cấu trúc phức tạp của văn xuôi ảnh hưởng đến kết quả thực nghiệm.

8.2.1.2. Kết quả theo từng tác phẩm

Bảng 8.2. Kết quả chi tiết của mô hình DBNet theo từng tác phẩm

Tên tác phẩm	Precision	Recall	F1-score
Lục Vân Tiên	1.0000	0.9991	0.9996
Truyện Kiều bản 1866	1.0000	1.0000	1.0000
Truyện Kiều bản 1871	1.0000	1.0000	1.0000
Truyện Kiều bản 1872	1.0000	0.9998	0.9999
ĐVS KTT-Quyển Thủ	0.9974	0.9822	0.9898
ĐVS KTT-Ngoại ký toàn thư	0.9997	0.9915	0.9956

ĐVSKTT-Bản kỷ toàn thư	0.9992	0.9945	0.9968
ĐVSKTT-Bản kỷ thực lục	0.9986	0.994	0.9963
ĐVSKTT-Bản kỷ tục biên	0.9988	0.9882	0.9934

Bảng 8.2 thể hiện kết quả của từng tác phẩm trong tập Validate của mô hình DBNet dựa trên các độ đo Precision, Recall, và F1-score. Một lần nữa, có thể thấy kết quả đạt được trên các tập thơ ở DBNet tốt hơn so với văn xuôi. Trong đó, các tập thơ là *Truyện Kiều* bản năm 1866 và 1871 cho kết quả rất ấn tượng với F1-score là 1.0. Còn đối với các ảnh văn xuôi, có thể thấy Bản kỷ của ĐVSKTT tuy chiếm số điểm dữ liệu lớn nhất trong bộ Quốc sử này nhưng mô hình lại cho kết quả tốt nhất so với các bản còn lại trong bộ với F1-score là 0.9968. Ngược lại, Quyển Thủ của ĐVSKTT chiếm tỉ lệ điểm dữ liệu ít nhất trong bộ nhưng lại có kết quả thấp nhất với F1-score là 0.9898, tuy vậy đây vẫn là một giá trị vẫn rất tốt. Nhìn chung, mô hình DBNet cho kết quả rất cao trên các ảnh thơ và văn xuôi.

Bảng 8.3. Kết quả chi tiết của mô hình EAST theo từng tác phẩm

Tên tác phẩm	Precision	Recall	F1-score
Lục Vân Tiên	0.9988	0.9997	0.9993
<i>Truyện Kiều</i> bản 1866	0.9997	0.9997	0.9997
<i>Truyện Kiều</i> bản 1871	0.9989	1.0000	0.9994
<i>Truyện Kiều</i> bản 1872	1.0000	0.9864	0.9931
ĐVSKTT-Quyển Thủ	0.9760	0.9531	0.9644
ĐVSKTT-Ngoại kỷ toàn thư	0.9730	0.9616	0.9672
ĐVSKTT-Bản kỷ toàn thư	0.9901	0.986	0.9880
ĐVSKTT-Bản kỷ thực lục	0.9929	0.9867	0.9898
ĐVSKTT-Bản kỷ tục biên	0.9889	0.9773	0.9831

Bảng 8.3 thể hiện kết quả của từng tác phẩm trong tập Validate của mô hình EAST dựa trên các độ đo Precision, Recall, và F1-score. Đối với mô hình này, cũng không là ngoại lệ khi có thể thấy kết quả đạt được trên các tập thơ tốt hơn so với

văn xuôi. Trong đó tập thơ Truyện Kiều bản 1866 cho ra kết quả cao nhất với F1-score là 0.9997. Với văn xuôi, Bản kỷ thực lục của bộ ĐVSKT có kết quả cao nhất với F1-score là 0.9898. Bên cạnh đó, Quyển Thủ cũng có kết quả thấp nhất với mô hình nay như với DBNet với F1-score là 0.9644. Nhìn chung thì mô hình EAST cho kết quả cao trên các ảnh là thơ và văn xuôi.

8.2.2. Kết quả bài toán Text Recognition

8.2.2.1. Kết quả giai đoạn Pre-training

Bảng 8.4. Kết quả Pre-training trên bộ Synthetic Nom String thuộc IHR-NomDB

Mô hình	Optimizer	seq_acc	char_acc	CER
Init-inject Captioner	Adam	0.1021	0.6927	0.2908
Attention-based Captioner	Adam	0.7759	0.9729	0.0252
CRNNxCTC	Adadelta	0.9150	0.9892	0.0105
CRNNxCTC + <i>Concat Connection</i>	Adadelta	0.9485	0.9936	0.0062
CNNxSeq2Seq	Adam	0.6854	0.9580	0.0401
CNNxSeq2Seq + <i>Concat Connection</i>	Adam	0.8449	0.9825	0.0168
CNNxTransformer	Adadelta	0.8726	0.9845	0.0133
CNNxTransformer + <i>Add Connection</i>	Adadelta	0.8895	0.9859	0.0115
CNNxTransformerDecoder	Adadelta	0.6414	0.9034	0.0620

Có thể thấy với tập Validate của bộ dữ liệu Synthetic Nom String, ngoài mô hình Init-inject Captioner có kết quả khá tệ với chỉ 0.1021 cho Độ chính xác ở mức chuỗi (seq_acc), ngay cả với một độ đo dễ dàng đạt giá trị cao như Độ chính xác ở mức ký tự (char_acc) thì mô hình này cũng chỉ cho ra kết quả khá với chỉ 0.6927, điều này hiển nhiên cũng dẫn đến một Tỷ lệ lỗi ký tự (CER) tương đối không tốt với 0.2908. Các mô hình còn lại đều có kết quả khá cao cho char_acc (đều > 0.9).

Về optimizer, các mô hình CRNNxCTC đạt kết quả tốt nhất khi sử dụng Adadelta tương tự như cách thử nghiệm được đề xuất trong bài báo gốc của nó [65]. Một điều đáng chú ý nữa là việc đề xuất thử nghiệm với Adadelta của chúng tôi cho các mô hình cần định thời learning rate một cách cẩn thận để có thể hội tụ như Transformer là một điều đúng đắn khi các mô hình này đạt kết quả khá tốt. Từ đó, khi huấn luyện các mô hình dựa trên Transformer, chúng tôi đã không cần phải sử dụng những cách định thời truyền thống hay cũng không cần phải quan tâm tới việc điều chỉnh learning rate.

Ngoài ra, với riêng 3 mô hình khả thi cho việc thử nghiệm các Kết nối tắt mà đã được đề cập ở [7.2.6](#) gồm CRNNxCTC, CNNxSeq2Seq, và CNNxTransformer, chúng tôi có thể dễ dàng thấy được sự hiệu quả của thử nghiệm này. Với phiên bản gốc (chưa thêm Kết nối tắt) của 3 mô hình này thì CRNNxCTC có kết quả tốt nhất nhưng không đáng kể so với CNNxTransformer và phải train khá lâu (hơn 5 giờ) để có được kết quả đó. Với phiên bản thử nghiệm các Kết nối tắt mà chúng tôi đề xuất, các mô hình này đều mang lại các kết quả tốt hơn so với phiên bản không sử dụng Kết nối tắt của chính nó. Trong đó, đáng chú ý nhất là mô hình CRNNxCTC sử dụng phép ghép (Concatenation) đã vượt trội (outperform) trên tất cả metric với seq_acc đạt 0.9485 và một char_acc gần như tuyệt đối là 0.9936, nên từ đó cũng dễ hiểu khi giá trị CER có được lại tốt như vậy với chỉ 0.0062. Gần 6% seq_acc bị bỏ qua có lẽ thuộc về các dạng câu có các ký tự chỉ xuất hiện một lần như chúng tôi đã đề cập trong [5.5](#). Mô hình CRNNxCTC cũng có cách biệt kết quả khá xa so với các mô hình còn lại và đây đồng thời cũng là mô hình tốt nhất cho giai đoạn Pre-training của chúng tôi với số params chỉ nhiều hơn phiên bản gốc của chính nó - cũng là mô hình có ít params nhất.

8.2.2.2. Kết quả Fine-tuning và Retraining

Từ các kết quả của giai đoạn Pre-training trong bảng [8.4](#), chúng tôi sẽ so sánh các mô hình trong 4 hướng tiếp cận của mục [6.3](#) và loại đi những mô hình thấp nhất nhưng việc so sánh trên sẽ không áp dụng giữa các mô hình được thử nghiệm

với Kết nối tắt và phiên bản gốc của chính nó. Cụ thể, chúng tôi sẽ loại đi 2 mô hình là Init-inject Captioner và CNNxTransformerDecoder, các mô hình còn lại sẽ được sử dụng cho cả 2 giai đoạn Fine-tuning và Retraining.

Bảng 8.5. Kết quả Fine-tuning và Retraining cho bộ dữ liệu NomNaOCR

Mô hình	Giai đoạn	seq_acc	char_acc	CER
Attention-based Captioner	Fine-tuning	0.1326	0.7680	0.2234
	Retraining	0.1024	0.7121	0.2714
CRNNxCTC	Fine-tuning	0.2627	0.8325	0.1617
	Retraining	0.2941	0.8473	0.1508
CRNNxCTC + <i>Concat Connection</i>	Fine-tuning	0.2266	0.8060	0.1823
	Retraining	0.2423	0.8189	0.1699
CNNxSeq2Seq	Fine-tuning	0.1095	0.7067	0.2711
	Retraining	0.1057	0.6995	0.2868
CNNxSeq2Seq + <i>Concat Connection</i>	Fine-tuning	0.1837	0.8176	0.1728
	Retraining	0.0979	0.6915	0.2815
CNNxTransformer	Fine-tuning	0.1963	0.7910	0.1771
	Retraining	0.1452	0.7013	0.2521
CNNxTransformer + <i>Add Connection</i>	Fine-tuning	0.2714	0.8490	0.1335
	Retraining	0.2423	0.8016	0.1664

Nhìn chung, tất cả mô hình trừ các mô hình CRNNxCTC, đều có kết quả tốt hơn khi thực hiện Fine-tuning trên các trọng số đã học được từ bộ dữ liệu Synthetic Nom String so với khi huấn luyện lại từ đầu. Bên cạnh đó, khi xem xét việc Fine-

tuning trên cho riêng 3 mô hình khả thi với các thử nghiệm Kết nối tắt thì một lần nữa đề xuất này của chúng tôi lại được củng cố khi các thử nghiệm này đều mang lại tốt hơn so với phiên bản gốc của chính nó nhưng cũng một lần nữa ngoại trừ CRNNxCTC. Chúng tôi nhận định với bộ dữ liệu NomNaOCR này, mô hình CRNNxCTC chỉ thực sự mang lại kết quả tốt khi được huấn luyện lại từ đầu và không thêm bất kỳ Kết nối tắt nào. Cụ thể với các cài đặt này, CRNNxCTC đã đạt độ chính xác theo mức câu cao nhất trong tất cả mô hình với 0.2941.

Một điều đáng chú ý khác là mô hình CNNxSeqSeq có kết quả khá tệ so với các mô hình khác khi được huấn luyện lại từ đầu với chỉ 0.0979 cho seq_acc và 0.6915 cho char_acc, kể cả khi đã sử dụng Kết nối tắt mà chúng tôi đề xuất, giá trị CER của mô hình vẫn không suy giảm và vẫn ở mức cao nhất so với các mô hình còn lại, lên tới 0.2868. Tuy nhiên, việc thực hiện Fine-tuning kết hợp với việc thử nghiệm Kết nối tắt đã cải thiện đáng kể hiệu suất mô hình như đã đề cập trong đoạn trên. CNNxSeq2Seq đồng thời cũng là kiến trúc có thời gian hội tụ khá lâu.

Với 3 độ đo mà chúng tôi sử dụng, ngoài CRNNxCTC có seq_acc cao nhất, mô hình có kết quả tốt nhất cho 2 giá trị còn lại là CNNxTransformer khi được Fine-tuning và sử dụng phép cộng để Kết nối tắt, với lần lượt 0.8490 cho char_acc và 0.1335 cho CER. Tuy nhiên, các metric này cũng không có sự chênh lệch đáng kể giữa 2 mô hình trên. Vì vậy, chúng tôi nhận định đây là 2 mô hình tốt nhất cho bài toán Text Recognition trên bộ dữ liệu NomNaOCR mà chúng tôi xây dựng.

8.2.2.3. Kết quả các ngưỡng 10 ký tự

Bộ dữ liệu Synthetic Nom String chúng tôi dùng để tiền huấn luyện có độ dài phổ biến trong khoảng 6 đến 8 ký tự và có độ dài lớn nhất là 10 ký tự, còn với bộ dữ liệu NomNaOCR mà chúng tôi xây dựng thì các khoảng phổ biến là 6, 8 và trên 17 ký tự và độ dài lớn nhất lên tới 24 ký tự. Vì vậy chúng tôi sẽ thực hiện đánh giá thêm sự tác động của độ dài ký tự lên các mô hình Recognition, bằng cách sử dụng một ngưỡng phân định là 10, chúng tôi sẽ xem xét kết quả các mô hình bị ảnh hưởng như thế nào với các đầu vào > 10 và ≤ 10 ký tự.

Bảng 8.6. Kết quả các ngưỡng 10 khi Fine-tuning và Retraining cho NomNaOCR

Mô hình	Giai đoạn	seq_acc		char_acc		CER	
		> 10	≤ 10	> 10	≤ 10	> 10	≤ 10
Attention-based Captioner	Fine-tuning	0.0369	0.2319	0.7713	0.7638	0.2225	0.2241
	Retraining	0.0119	0.1966	0.7078	0.7256	0.2823	0.2592
CRNNxCTC	Fine-tuning	0.1456	0.3843	0.8401	0.8094	0.1332	0.1913
	Retraining	0.1497	0.4440	0.8520	0.8328	0.1317	0.1707
CRNNxCTC + <i>Concat Connection</i>	Fine-tuning	0.1026	0.3552	0.8104	0.7926	0.1557	0.2099
	Retraining	0.1153	0.3741	0.8237	0.8041	0.1411	0.1999
CNNxSeq2Seq	Fine-tuning	0.0067	0.2160	0.6913	0.7486	0.3014	0.2389
	Retraining	0.0143	0.2009	0.6957	0.7116	0.2991	0.2729
CNNxSeq2Seq + <i>Concat Connection</i>	Fine-tuning	0.0617	0.3106	0.8177	0.8184	0.1727	0.1730
	Retraining	0.0101	0.1888	0.6798	0.7235	0.3003	0.2613
CNNxTransformer	Fine-tuning	0.0192	0.3803	0.7759	0.8311	0.1957	0.1576
	Retraining	0.0322	0.2621	0.6951	0.7195	0.2522	0.2515
CNNxTransformer + <i>Add Connection</i>	Fine-tuning	0.1393	0.4084	0.8473	0.8545	0.1299	0.1375
	Retraining	0.0859	0.4046	0.7913	0.8293	0.1753	0.1572

Có thể thấy các mô hình đều có seq_acc tốt hơn rất nhiều khi dự đoán trên các đầu vào ngắn hơn, ở đây là từ 10 ký tự trở xuống. Điều này cũng dễ hiểu với một độ đo dễ bị mắc sai làm như seq_acc vì chỉ cần dự đoán sai một ví trí thì tất cả ký tự khác trong chuỗi đều bị tính là sai. Do vậy, việc đánh giá kết quả của một mô hình bị ảnh hưởng bởi độ dài ký tự bằng seq_acc sẽ không mang lại nhiều ý nghĩa.

Với các đầu vào > 10 ký tự, mô hình CNNxSeq2Seq khi được Fine-tuning dường như không dự đoán được chính xác một chuỗi nào với giá trị seq_acc chỉ có 0.0067, thấp nhất trong tất cả mô hình. Với các đầu vào ≤ 10 ký tự, CNNxSeq2Seq có kết quả khá thấp khi được huấn luyện lại từ đầu, dù đã có sử dụng thêm Kết nối tắt mà chúng tôi đề xuất với chỉ 0.1888 cho seq_acc. Bên cạnh đó, mô hình có kết quả tốt nhất theo độ đo này vẫn là CRNNxCTC khi được huấn luyện lại, tương tự như đã thấy ở bảng 8.5, với seq_acc lần lượt là 0.1497 và 0.4440 cho đầu vào > 10 và ≤ 10 ký tự.

Còn về char_acc, các mô hình đa phần đều có kết quả tốt hơn khi dự đoán các chuỗi ngắn hay từ 10 ký tự trở xuống trừ 2 mô hình Attention-based Captioner với cách biệt không đáng kể so với đầu vào > 10 ký tự và CRNNxCTC. Và với riêng các mô hình CRNNxCTC, đáng ngạc nhiên là kết quả trên các chuỗi dài hay > 10 ký tự lại tốt hơn nhiều trên các chuỗi ngắn hay ≤ 10 ký tự với giá trị char_acc cao nhất là 0.8520 khi thực hiện Retraining. Với các đầu vào ≤ 10 ký tự thì mô hình CNNxTransformer kết hợp với việc Fine-tuning và sử dụng Kết nối tắt mà chúng tôi đề xuất mới là mô hình tốt nhất với 0.8545 cho char_acc, bên cạnh mô hình có kết quả thấp nhất là CNNxSeq2Seq khi được huấn luyện lại và chỉ đạt được 0.7116 cho char_acc. Dù đã được thử nghiệm với Kết nối Tắt, mô hình này vẫn có giá trị char_acc khá tệ cho các đầu vào > 10 ký tự với chỉ 0.6798.

Trên thang đo CER, mô hình CNNxTransformer khi được Fine-tuning và thử nghiệm với Kết nối tắt vẫn là mô hình có hiệu quả nhất khi cả 2 giá trị CER cho 2 kiểu đầu vào dài và ngắn đều đạt được kết quả cao nhất trong tất cả mô hình. Cụ thể, mô hình này đạt giá trị CER là 0.1299 trên các chuỗi dài hay > 10 ký tự, còn tốt hơn cả khi dự đoán cho các chuỗi ngắn ≤ 10 ký tự với giá trị CER đạt 0.1375 và cũng như trên thang đo char_acc, mô hình thấp nhất với loại đầu vào ≤ 10 ký tự này vẫn là CNNxSeq2Seq khi được huấn luyện lại với giá trị CER khá lớn là 0.2729. Ngay cả khi được Fine-tuning, mô hình này vẫn có kết quả cho các đầu vào > 10 ký tự khá tệ so với các mô hình khác khi có giá trị CER khá cao là 0.3014.

8.2.3. Kết quả End-to-End

Trong phần này, chúng tôi triển khai đánh giá cho sự kết hợp của 2 bài toán phát hiện và nhận dạng các ký tự Hán-Nôm trong ảnh dựa trên tập Validate của bộ dữ liệu NomNaOCR. Để thực hiện đánh giá này chúng tôi tiến hành đưa các Page vào các mô hình phát hiện văn bản là DBNet và EAST để từ đó thu được đầu ra là các Patch. Sau đó, chúng tôi sẽ dùng các Patch này làm đầu vào cho 4 mô hình tốt nhất tương ứng với 4 cách tiếp cận của chúng tôi cho bài toán Recognition. Cuối cùng chúng tôi sẽ có được các nhãn đầu ra là các bounding box và các chuỗi ký tự tương ứng trong mỗi box. Vì đầu vào của cách đánh giá này là Page tương tự như của bài toán Text Detection nên chúng tôi cũng sẽ triển khai chiến lược đánh giá theo 3 hướng bao gồm Toàn bộ, Thơ, và Văn xuôi cho các mô hình kết hợp sử dụng các metric đánh giá End-to-End đã đề cập trong [8.1.1](#).

8.2.3.1. Kết quả trên toàn bộ ảnh

Bảng 8.7. Kết quả End-to-End trên toàn bộ ảnh của tập Validate

Mô hình kết hợp		Finetune	Precision	Recall	F1-score
DBNet	Attention-based Captioner	✓	0.7750	0.7741	0.7745
	CRNNxCTC		0.9239	0.9226	0.9232
	CNNxSeq2Seq + <i>Concat Connection</i>	✓	0.8524	0.8470	0.8497
	CNNxTransformer + <i>Add Connection</i>	✓	0.9035	0.9032	0.9034
EAST	Attention-based Captioner	✓	0.7077	0.7144	0.7110
	CRNNxCTC		0.8735	0.8755	0.8745
	CNNxSeq2Seq + <i>Concat Connection</i>	✓	0.7886	0.7981	0.7933
	CNNxTransformer + <i>Add Connection</i>	✓	0.8519	0.8603	0.8561

Có thể thấy, với bộ dữ liệu NomNaOCR các mô hình có sự kết hợp của DBNet cho kết quả tốt hơn nhiều so với sự kết hợp của EAST, khoảng 3-5% F1-score. Để thấy, mô hình CRNNxCTC có kết quả tốt nhất trong các bộ kết hợp với mô hình EAST với F1-score là 0.8745. Tuy nhiên, khi kết hợp với DBNet, mô hình này cho kết quả vượt trội trong tất cả mô hình kết hợp với F1-score đạt 0.9232. Bên cạnh đó, các kết hợp có sự tham gia của mô hình Attention-based Captioner khi Fine-tuning có kết quả tệ nhất với F1-score lần lượt là 0.7745 (DBNet) và 0.7110 (EAST). Ngoài ra, cũng dễ thấy các sự kết hợp khác giữa: DBNet/EAST và CNNxTransformer khi được Fine-tuning cùng với sử dụng Kết nối tắt mà chúng tôi đề xuất, đều cho kết quả khá tốt khi có F1-score đều trên 0.85.

8.2.3.2. Kết quả chi tiết trên thơ và văn xuôi

Bảng 8.8. Kết quả End-to-End trên các ảnh thơ của tập Validate

Mô hình kết hợp		Finetune	Precision	Recall	F1-score
DBNet	Attention-based Captioner	✓	0.7070	0.7040	0.7055
	CRNNxCTC		0.9396	0.9396	0.9396
	CNNxSeq2Seq + <i>Concat Connection</i>	✓	0.8632	0.8610	0.8620
	CNNxTransformer + <i>Add Connection</i>	✓	0.9256	0.9252	0.9254
EAST	Attention-based Captioner	✓	0.7786	0.7809	0.7798
	CRNNxCTC		0.9214	0.9213	0.9214
	CNNxSeq2Seq + <i>Concat Connection</i>	✓	0.8432	0.8462	0.8447
	CNNxTransformer + <i>Add Connection</i>	✓	0.9087	0.9120	0.9104

Tương tự với kết quả tổng thể trên toàn bộ ảnh, các mô hình có sự tham gia của DBNet đa phần có kết quả tốt hơn so với của EAST. Và các kết hợp có sự tham gia của mô hình Attention-based Captioner khi được Fine-tuning cũng cho kết quả

thấp nhất với F1-score lần lượt là 0.7055 (DBNet) và 0.7798 (EAST). Nhìn chung khi đánh giá các ảnh thơ trên tập Validate của NomNaOCR, các kết hợp gồm: DBNet/EAST và CRNNxCTC, DBNet/EAST và CNNxTransformer (Fine-tuning và Kết nối tắt) đều đạt kết quả khá tốt khi trên 0.9. Đặc biệt là kết hợp giữa DBNet và CRNNxCTC có kết quả vượt trội so với các kết hợp còn lại khi F1-score đạt được lên tới 0.9396.

Bảng 8.9. Kết quả End-to-End trên các ảnh văn xuôi của tập Validate

Mô hình kết hợp		Finetune	Precision	Recall	F1-score
DBNet	Attention-based Captioner	✓	0.8026	0.7905	0.7965
	CRNNxCTC		0.9206	0.9190	0.9198
	CNNxSeq2Seq + <i>Concat Connection</i>	✓	0.8401	0.8485	0.8442
	CNNxTransformer + <i>Add Connection</i>	✓	0.8989	0.8987	0.8988
EAST	Attention-based Captioner	✓	0.6922	0.6997	0.6959
	CRNNxCTC		0.8630	0.8653	0.8641
	CNNxSeq2Seq + <i>Concat Connection</i>	✓	0.7766	0.7875	0.7820
	CNNxTransformer + <i>Add Connection</i>	✓	0.8395	0.8489	0.8442

Cuối cùng, với văn xuôi, có thể thấy các mô hình có sự tham gia của DBNet vượt trội hoàn toàn so với EAST, những chênh lệch này ở mức khá cao, từ 6% đến 10%. Còn các kết hợp có sự tham gia của CRNNxCTC cũng đạt kết quả tốt nhất với F1-score lần lượt là 0.9198 (DBNet) và 0.8641 (EAST). Và các kết hợp có sử dụng mô hình Attention-based Captioner khi Fine-tuning vẫn là thấp nhất với F1-score lần lượt là 0.7965 (DBNet) và 0.6959 (EAST). Ngoài ra, các mô hình kết hợp gồm DBNet/EAST và CRNNxCTC, DBNet/EAST và CNNxTransformer (Fine-tuning và Kết nối tắt) đều đạt kết quả cao khi đều trên 0.8.

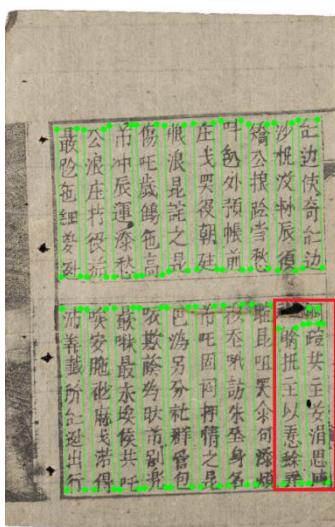
8.2.3.3. Nhận định

Mô hình CNNxTransformer được Fine-tuning và sử dụng Kết nối tắt mà chúng tôi đề xuất khi đánh giá trên tập Validate của bộ dữ liệu NomNaOCR dành riêng cho nhiệm vụ nhận dạng thì có kết quả không chêch lệch nhiều và đa phần tốt hơn so với CRNNxCTC như đã thấy trong 8.2.2.2, nhưng khi kết hợp với các mô hình Detection thì CRNNxCTC lại có kết quả tốt hơn. Đặc biệt là với sự kết hợp giữa DBNet và CRNNxCTC tuy không chêch lệch nhiều so với sự kết hợp giữa DBNet và CNNxTransformer (Fine-tuning và Kết nối tắt) nhưng lại có kết quả vượt trội hoàn toàn so với các mô hình kết hợp còn lại. Với kết quả này, có thể thấy mô hình CRNNxCTC phù hợp hơn với môi trường thực tế khi các bounding box cho việc nhận dạng không được đẹp và chính xác như khi được gán nhãn.

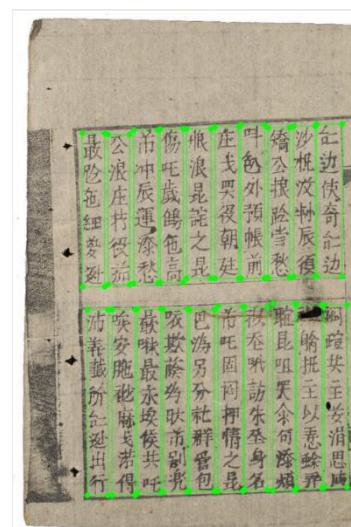
8.3. Phân tích lỗi

8.3.1. Phân tích lỗi cho bài toán Text Detection

Đối với các tập thơ thì DBNet là mô hình tốt nhất của chúng tôi, phát hiện gần như chính xác hoàn toàn, chỉ đúng một trường hợp trong hình 8.3 (khung màu đỏ), bị sai khi xác định câu thứ hai trong tác phẩm Lục Vân Tiên. Điều này xảy ra là do Patch bị mờ và ảnh hưởng bởi vết mực lớn.



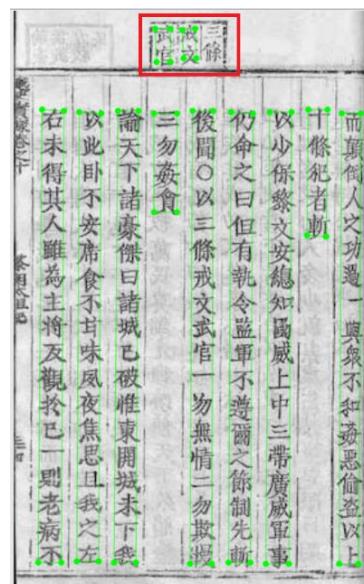
Hình 8.1. Ví dụ phát hiện sai trên tập thơ
của mô hình DBNet



Hình 8.2. Ví dụ phát hiện đúng hoàn toàn
của mô hình DBNet

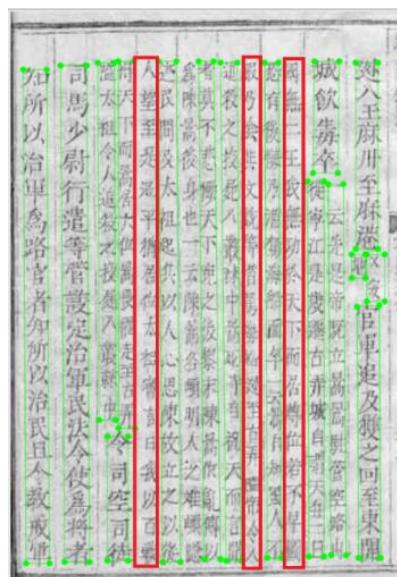
Đối với các tác phẩm văn xuôi, DBNet thường phát hiện sai ở 3 trường hợp sau:

- Mô hình thường nhầm lẫn với các văn bản nằm ngoài nội dung của tác phẩm như hình 8.5. Vì các Patch này không những có hình dạng giống với các Patch ở phần nội dung chính mà còn có kích thước các ký tự khá tương đồng.



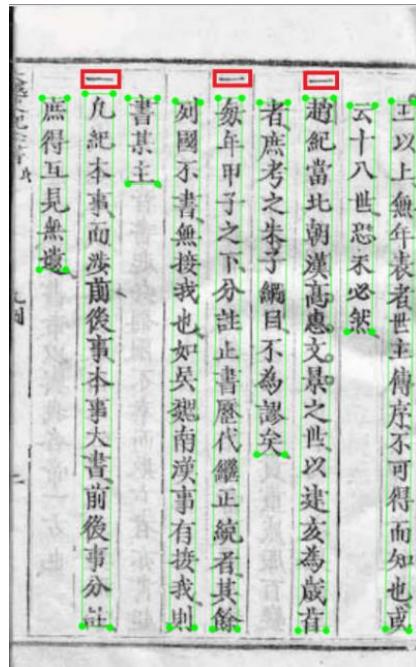
Hình 8.3. Ví dụ phát hiện các văn bản nằm ngoài nội dung

- Các ký tự trong những Patch rất nhỏ đồng thời nằm sát vào nhau như hình 8.6 cũng khiến mô hình nhầm lẫn dẫn đến phát hiện bị sót. Các ký tự rất nhỏ này sẽ dẫn đến việc mất một số nét chữ do độ phân giải bị giảm sút.



Hình 8.4. Ví dụ phát hiện thiếu sót các văn bản

- Các ký tự “—” xuất hiện ở vị trí đầu tiên trong patch (hình 8.7), thì mô hình thường xuyên không phát hiện được. Điều này xảy ra do mô hình không phân biệt được box bao quanh toàn bộ văn bản và chữ “—”.

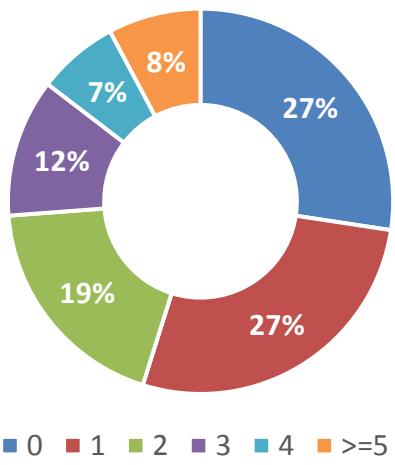


Hình 8.5. Ví dụ không phát hiện được ký tự “—” ở đầu patch

8.3.2. Phân tích lỗi cho bài toán Text Recognition

Với mô hình CNNxTransformer được Fine-tuning sử dụng phép cộng để kết nối tắt, chúng tôi đã được kết quả tốt trên nhiều độ đo cho riêng bài toán Recognition. Mô hình này hầu như chỉ dự đoán sai từ 1 đến 2 ký tự trong mỗi Patch, đã được thống kê trong hình 8.8.

Ngoài ra, từ bảng 5.3 có thể thấy, đa phần số lượng ký tự trong mỗi Patch là từ 17 trở lên. Cả 2 điều trên cũng giải thích tại sao trong khi CER có kết quả khá tốt nhưng Sequence Accuracy lại rất thấp, như đã thấy trong 8.2.2.2.



Hình 8.6. Phân phối các ký tự dự đoán sai của tập Validate

行
 師
 討
 鋤
 諸
 處
 蓮
 大
 下
 賴
 以
 安
 平
 曹
 三
 十
 年
 不
 意
 (1)
 (2)
 (3)
 (4)

Patch Dự đoán Nhãn

sai của tập Validate

Hình 8.7. Ví dụ các dự đoán sai

Ngoài ra, chỉ có số lượng rất ít các Patch bị dự đoán sai trên 3 ký tự. Hình 8.9 cho ta thấy một ví dụ về các lỗi xuất hiện trong dự đoán khi đánh giá trên một Patch của Bản kỷ tục biên của ĐVSHTT. Có thể thấy rằng các trường hợp dự đoán sai (1), (2), (3) và (4) có hình dạng rất giống với các ký tự trong nhãn đúng cũng như là trong ảnh. Thậm chí với trường hợp (3), nếu nhìn thoáng qua có thể khiến người bình thường nhầm lẫn giữa 2 ký tự này. Điều này có thể xảy ra do sự thay đổi trong cách viết giữa các ký tự (trường hợp 2, 3).

Bên cạnh đó, chúng tôi cũng thực hiện thống kê số lượng các ký tự xuất hiện ít (từ 1 đến 3 lần) trong tập Train nhưng mô hình vẫn dự đoán chính xác trên tập Validate, được thể hiện qua bảng 8.10. Cụ thể, trong tập Validate có 764 ký tự ít xuất hiện trong tập Train (xuất hiện từ 1 đến 3 lần) thì mô hình dự đoán được tổng cộng 187 ký tự tương đương 24.48%.

Bảng 8.10. Số ký tự vẫn được dự đoán đúng dù ít xuất hiện trong tập Train

Tần suất xuất hiện	Số lượng
1	32
2	70
3	85

Chương 9. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

9.1. Tổng kết

Trong khóa luận này, chúng tôi đã đạt được những mục tiêu được đề ra trong 1.3 bằng việc xây dựng hoàn thiện một bộ dữ liệu tốt và lớn nhất Việt Nam hiện tại dành cho chữ Hán-Nôm với tên NomNaOCR, gồm 2953 Page là các ảnh scan của các trang văn bản cũ và 38318 Patch được trích xuất từ các Page này cùng các chuỗi ký tự Hán-Nôm kỹ thuật số tương ứng, nhằm phục vụ cho 2 bài toán phát hiện và nhận dạng các ký tự Hán-Nôm viết tay theo mức chuỗi.

Cùng với đó, chúng tôi cũng đã triển khai thành công các mô hình Deep Learning giải quyết 2 bài toán trên theo nhiều hướng tiếp cận và cũng đã đạt được một số thành tựu nhất định khi có kết quả rất cao trên tập Validate của bộ dữ liệu NomNaOCR cho bài toán Detection với 2 mô hình EAST và DBNet đều đạt được hơn 0.98 điểm cho cả 3 độ đo Precision, Recall, F1-score và trên cả 3 hướng đánh giá gồm Toàn bộ ảnh, Thư, hay Văn xuôi.

Ngoài ra, với 4 hướng tiếp cận được đề xuất cho bài toán Recognition bên cạnh việc thực hiện thử nghiệm trên 3 giai đoạn Pre-training, Fine-tuning và Retraining, chúng tôi cũng đã có được những kết quả rất tích cực với giá trị Character Accuracy đều đạt trên 84% cho cả 2 mô hình CRNNxCTC khi được huấn luyện lại và CNNxTransformer khi được Fine-tuning cùng với một Kết nối tắt mà chúng tôi đề xuất, đây đồng thời cũng là mô hình có giá trị CER nhỏ nhất mà chúng tôi đạt được trên tập Validate của bài toán Recognition với 13.35%.

Hơn thế nữa, chúng tôi cũng thực hiện đánh giá kết hợp (End-to-End) cho 2 bài toán trên và nhận được một kết quả khá tốt khi sự kết hợp giữa DBNet và CRNNxCTC tuy không chêch lệch nhiều so với sự kết hợp giữa DBNet và CNNxTransformer (Fine-tuning và Kết nối tắt) nhưng lại có kết quả vượt trội hoàn toàn so với các mô hình kết hợp còn lại với F1-score đạt 92.32%. Ngoài ra, nhiều phân tích chi tiết khác như kết quả nhận dạng trên các ngưỡng 10 ký tự hay các phân tích lỗi cho các mô hình tốt nhất cũng đã được chúng tôi đưa ra và làm rõ.

9.2. Hướng nghiên cứu trong tương lai

Dựa vào các kết quả đã đạt được, có thể thấy với riêng bài toán Text Detection, 2 mô hình được sử dụng là EAST và DBNet đều đã đạt giá trị rất tốt với trên 98% cho cả 3 độ đo Precision, Recall và F1-score. Vì vậy, trong tương lai chúng tôi sẽ tập trung cải thiện cho riêng bài toán Text Recognition. Từ những kết quả có được từ bài toán này, chúng tôi nhận thấy được hiệu suất rất khả quan trên bộ dữ liệu NomNaOCR của 2 mô hình CRNNxCTC và CNNxTransformer (Fine-tuning và Kết nối tắt), do đó chúng tôi sẽ nghiên cứu thêm cách kết hợp 2 mô hình này lại bằng việc huấn luyện CNNxTransformer theo 3 giai đoạn đã đề cập trong [7.2.1](#) như bình thường nhưng mô hình sẽ được tối ưu bằng CTC Loss như CRNN. Hay xa hơn nữa là xây dựng một mô hình End-to-End như STN-OCR [73], có thể huấn luyện được cùng lúc cả 2 nhiệm vụ phát hiện và nhận dạng văn bản.

Bên cạnh đó, việc phát sinh chuỗi hiện tại của tất cả mô hình đều chỉ đơn giản là sử dụng Tìm kiếm tham lam (Greedy search) để tìm các token có xác suất có điều kiện cao nhất tại mỗi timestep. Tuy nhiên, với cách làm này, chuỗi được phát sinh có thể không phải là chuỗi có xác suất cao nhất. Điều này có thể được giải quyết bằng Tìm kiếm Vết cạn (Exhaustive Search), tức kiểm tra tất cả các chuỗi đầu ra có thể và trả về chuỗi có xác suất có điều kiện cao nhất, nhưng chi phí tính toán cho giải thuật này là quá lớn. Vì vậy, trong tương lai chúng tôi có thể sẽ sử dụng một thuật toán cải tiến hơn là Tìm kiếm Chùm (Beam search) để cân bằng giữa chi phí tính toán và chất lượng tìm kiếm bằng cách giữ lại N chuỗi có xác suất lớn theo đường đi tốt nhất, cuối cùng chuỗi được chọn sẽ là chuỗi có xác suất cao nhất trong N chuỗi đó. Hay xa hơn nữa là tích hợp mô hình ngôn ngữ (language model) để giải mã đầu ra theo ngữ cảnh của bài toán Text Recognition.

Ngoài ra, bộ dữ liệu NomNaOCR đã xây dựng cũng sẽ được chúng tôi tăng thêm về số lượng bằng nhiều tác phẩm lịch sử khác của Việt Nam và từ những gì thu thập được, chúng tôi sẽ mở rộng bộ dữ liệu này cho nhiều bài toán khác ngoài OCR như dịch các nội dung Hán-Nôm sang Quốc Ngữ, ...

TÀI LIỆU THAM KHẢO

- [1] B. John , C. Lee , L. Stephen , P. John , S. D. Neil and N. T. Việt, "Vietnamese Nôm Preservation Foundation," [Online]. Available: <http://www.nomfoundation.org>.
- [2] M.-S. Kim, M.-D. Jang, H.-I. Choi, T.-H. Rhee, J.-H. Kim and H.-K. Kwag, "Digitalizing scheme of handwritten Hanja historical documents," in First International Workshop on Document Image Analysis for Libraries, 2004. Proceedings., 2004, pp. 321-327.
- [3] C. L. Liu and Y. Lu, Advances in Chinese Document and Text Processing, vol. 02, World Scientific, 2017.
- [4] T. Phan, K. Nguyen and M. Nakagawa, "A Nom Historical Document Recognition System for Digital Archiving," Int. J. Doc. Anal. Recognit., vol. 19, p. 49–64, 2016.
- [5] T. Van Phan, B. Zhu and M. Nakagawa, "Development of Nom Character Segmentation for Collecting Patterns from Historical Document Pages," in Proceedings of the 2011 Workshop on Historical Document Imaging and Processing, 2011, p. 133–139.
- [6] A. D. Le, D. Mochihashi, K. Masuda, H. Mima and N. T. Ly, "Recognition of Japanese historical text lines by an attention-based encoder-decoder and text line generation," in Proceedings of the 5th International Workshop on Historical Document Imaging and Processing, 2019, p. 37–41.
- [7] E. Granell, E. Chammas, L. Likforman-Sulem, C.-D. Martínez-Hinarejos, C. Mokbel and B.-I. Cirstea, "Transcription of Spanish Historical Handwritten Documents with Deep Neural Networks," Journal of Imaging, vol. 4, p. 15, 2018.
- [8] E. Chammas, C. Mokbel and L. Likforman-Sulem, "Handwriting Recognition of Historical Documents with Few Labeled Data," in 2018 13th IAPR

- International Workshop on Document Analysis Systems, 2018, pp. 43-48.
- [9] M. T. Vu, V. L. Le and M. Beurton-Aimar, "IHR-NomDB: The Old Degraded Vietnamese Handwritten Script Archive Database," in Document Analysis and Recognition - ICDAR 2021, 2021, pp. 85-99.
 - [10] K. Nguyen, C. Nguyen and M. Nakagawa, "Nom Document Digitalization by Deep Convolution Neural Networks," Pattern Recognition Letters, 2020.
 - [11] C. K. Nguyen, C. T. Nguyen and N. Masaki, "Tens of Thousands of Nom Character Recognition by Deep Convolution Neural Networks," in Proceedings of the 4th International Workshop on Historical Document Imaging and Processing, 2017, p. 37–41.
 - [12] S. Anuj, "Building Custom Deep Learning Based OCR models," 2022. [Online]. Available: <https://nanonets.com/blog/attention-ocr-for-text-recognition>. [Accessed 20 June 2022].
 - [13] S. Iryna, "OCR Algorithms: Digitization of the Business Processes," 1 December 2020. [Online]. Available: <https://labelyourdata.com/articles/automation-with-ocr-algorithm>. [Accessed 20 June 2022].
 - [14] S. Iryna, "The Era of Digitization: Why Do the Automated Data Collection Systems Matter?," 23 November 2020. [Online]. Available: <https://labelyourdata.com/articles/automated-data-collection>. [Accessed 20 June 2022].
 - [15] A. Rahul, "Deep Learning Based OCR for Text in the Wild," 2022. [Online]. Available: <https://nanonets.com/blog/deep-learning-ocr>. [Accessed 20 June 2022].
 - [16] L. Tung, 25 July 2021. [Online]. Available: <http://tutorials.aiclub.cs.uit.edu.vn/index.php/2021/07/25/ai-tempo-run-gioi-thieu-bai-toan-scene-text>. [Accessed 20 June 2022].

- [17] S. Gidi, "A gentle introduction to OCR," 22 October 2018. [Online]. Available: <https://towardsdatascience.com/a-gentle-introduction-to-ocr-ee1469a201aa>. [Accessed 20 June 2022].
- [18] N. Yuval, W. Tao, C. Adam, B. Alessandro, W. Bo and Y. N. Andrew, "Reading Digits in Natural Images with Unsupervised Feature Learning," in NIPS Workshop on Deep Learning and Unsupervised Feature Learning, 2011.
- [19] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. i. Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazàn and L. P. de las Heras, "ICDAR 2013 Robust Reading Competition," in 2013 12th International Conference on Document Analysis and Recognition, 2013, pp. 1484-1493.
- [20] D. Karatzas, L. a. N. A. Gomez-Bigorda, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu, F. Shafait, S. Uchida and E. Valveny, "ICDAR 2015 competition on Robust Reading," in 2015 13th International Conference on Document Analysis and Recognition, 2015, pp. 1156-1160.
- [21] C. K. Ch'ng and C. S. Chan, "Total-Text: A Comprehensive Dataset for Scene Text Detection and Recognition," in 2017 14th IAPR International Conference on Document Analysis and Recognition, vol. 1, 2017, pp. 935-942.
- [22] R. Smith, "An Overview of the Tesseract OCR Engine," in Ninth International Conference on Document Analysis and Recognition, vol. 2, 2007, pp. 629-633.
- [23] G. Ankush, V. Andrea and Z. Andrew, "Synthetic Data for Text Localisation in Natural Images," in IEEE Conference on Computer Vision and Pattern Recognition, 2016.
- [24] L. Chenxia, L. Weiwei, G. Ruoyu, Y. Xiaoting, J. Kaitao, D. Yongkun, D. Yuning, Z. Lingfeng, L. Baohua, H. Xiaoguang, Y. Dianhai and M. Yanjun, "PP-OCRV3: More Attempts for the Improvement of Ultra Lightweight OCR System," arXiv:2206.03001v2, 2022.
- [25] A. Zhang, Z. C. Lipton, M. Li and A. J. Smola, "Convolutional Neural

- Networks," in Dive into Deep Learning, 2022, p. 225.
- [26] P. B. C. Quoc, "Tìm Hiểu Convolutional Neural Networks Cho Phân Loại Ảnh," 3 April 2019. [Online]. Available: <https://pbcquoc.github.io/cnn>. [Accessed 14 June 2022].
- [27] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol. 86, pp. 2278-2324, 1998.
- [28] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in Proceedings of the 32nd International Conference on Machine Learning, vol. 37, 2015.
- [29] A. Zhang, Z. C. Lipton, M. Li and A. J. Smola, "Batch Normalization," in Dive into Deep Learning, 2022, pp. 279-280.
- [30] T. Sivaram, "Skip Connections | All You Need to Know About Skip Connections," 24 August 2021. [Online]. Available: <https://www.analyticsvidhya.com/blog/2021/08/all-you-need-to-know-about-skip-connections>. [Accessed 15 June 2022].
- [31] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778.
- [32] G. Huang, Z. Liu, L. Van Der Maaten and K. Q. Weinberger, "Densely Connected Convolutional Networks," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2261-2269.
- [33] C. N. M. Anil, "A 2021 guide to Semantic Segmentation," 2021. [Online]. Available: <https://nanonets.com/blog/semantic-image-segmentation-2020>. [Accessed 15 June 2022].
- [34] L. Fei-Fei, J. Justin and Y. Serena , "Lecture 11: Detection and Segmentation," 10 May 2017. [Online]. Available:

http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture11.pdf. [Accessed 15 June 2022].

- [35] J. Long, E. Shelhamer and T. Darrell, "Fully convolutional networks for semantic segmentation," in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3431-3440.
- [36] O. Ronneberger, F. Philipp and B. Thomas , "U-Net: Convolutional Networks for Biomedical Image Segmentation," in Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, 2015, p. 234–241.
- [37] C. Olah, "Understanding LSTM Networks," 27 August 2015. [Online]. Available: <https://colah.github.io/posts/2015-08-Understanding-LSTMs>. [Accessed 16 June 2022].
- [38] TensorFlow Authors, "Word embeddings," 16 January 2022. [Online]. Available: https://www.tensorflow.org/text/guide/word_embeddings. [Accessed 16 June 2022].
- [39] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, pp. 1735-1780, 1997.
- [40] A. Zhang, Z. C. Lipton, M. Li and A. J. Smola, "Long Short-Term Memory (LSTM)," in Dive into Deep Learning, 2022, pp. 354-355.
- [41] K. Raimi, “Animated RNN, LSTM and GRU,” 14 December 2018. [Trực tuyến]. Available: <https://towardsdatascience.com/animated-rnn-lstm-and-gru-ef124d06cf45>. [Đã truy cập 11 July 2022].
- [42] K. Cho, B. v. Merriënboer, D. Bahdanau and Y. Bengio, "On the Properties of Neural Machine Translation: Encoder–Decoder Approaches," Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, pp. 103-111, 2014.
- [43] A. Zhang, Z. C. Lipton, M. Li and A. J. Smola, "Gated Recurrent Units (GRU)," in Dive into Deep Learning, 2022, pp. 347-351.

- [44] I. Sutskever, O. Vinyals and V. L. Quoc, "Sequence to Sequence Learning with Neural Networks," in Proceedings of the 27th International Conference on Neural Information Processing Systems, vol. 2, 2014, p. 3104–3112.
- [45] K. Cho, B. v. Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk and Y. Bengio, "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation," in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, 2014, pp. 1724-1734.
- [46] A. Jay, "Visualizing A Neural Machine Translation Model (Mechanics of Seq2seq Models With Attention)," 9 May 2018. [Online]. Available: <https://jalammar.github.io/visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-attention>. [Accessed 17 June 2022].
- [47] D. Bahdanau, K. H. Cho and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," arXiv preprint arXiv:1409.0473, 2014.
- [48] american_express, "A Comprehensive Guide to Attention Mechanism in Deep Learning for Everyone," 20 November 2019. [Online]. Available: <https://www.analyticsvidhya.com/blog/2019/11/comprehensive-guide-attention-mechanism-deep-learning>. [Accessed 17 June 2022].
- [49] A. Zhang, Z. C. Lipton, M. Li and A. J. Smola, "Attention Scoring Functions," in Dive into Deep Learning, 2022, pp. 405-406.
- [50] P. B. C. Quoc, "Tìm hiểu mô hình Transformer," 20 March 2020. [Online]. Available: <https://pbcquoc.github.io/transformer>. [Accessed 18 June 2022].
- [51] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin, "Attention is All You Need," in Advances in Neural Information Processing Systems, 2017, pp. 5998-6008.
- [52] L. Minh-Thang, P. Hieu and C. D. Manning, "Effective Approaches to Attention-based Neural Machine Translation," in Proceedings of the 2015

- Conference on Empirical Methods in Natural Language Processing, 2015, pp. 1412-1421.
- [53] T. Authors, "Transformer model for language understanding," 31 May 2022. [Online]. Available: <https://www.tensorflow.org/text/tutorials/transformer>. [Accessed 18 June 2022].
- [54] A. Zhang, Z. C. Lipton, M. Li and A. J. Smola, "Transformer," in Dive into Deep Learning, 2022, pp. 425-427.
- [55] A. Team. [Online]. Available: <https://www.automa.site>. [Accessed 29 June 2022].
- [56] M. Liao, Y. Wan, C. Yao, K. Chen and X. Bai, "Real-Time Scene Text Detection with Differentiable Binarization," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 11474-11481, 2020.
- [57] Y. Baek, D. Nam, S. Park, J. Lee, S. Shin, J. Baek, C. Y. Lee and H. Lee, "CLEval: Character-Level Evaluation for Text Detection and Recognition Tasks," in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2020, pp. 2404-2412.
- [58] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama and K. Murphy, "Speed/Accuracy Trade-Offs for Modern Convolutional Object Detectors," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 3296-3297.
- [59] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He and J. Liang, "EAST: An Efficient and Accurate Scene Text Detector," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2642-2651.
- [60] K. Wang, B. Babenko và S. Belongie, "End-to-end scene text recognition," trong 2011 International Conference on Computer Vision, 2011, pp. 1457-1464.
- [61] ROIS-DS Center for Open Data in the Humanities, "Kuzushiji Recognition,"

2019. [Online]. Available: <https://www.kaggle.com/competitions/kuzushiji-recognition>.
- [62] M. Tanti, A. Gatt and K. P. Camilleri, "Where to put the image in an image caption generator," *Natural Language Engineering*, vol. 24, p. 467–489, 2018.
- [63] Z. Wojna, A. N. Gorban, D.-S. Lee, K. Murphy, Q. Yu, Y. Li and J. Ibarz, "Attention-Based Extraction of Structured Information from Street View Imagery," in 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), vol. 1, 2017, pp. 844-850.
- [64] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel and Y. Bengio, "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention," in Proceedings of the 32nd International Conference on International Conference on Machine Learning, vol. 37, 2015, p. 2048–2057.
- [65] B. Shi, X. Bai and C. Yao, "An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 2298-2304, 2017.
- [66] K. S. Atul and K. Pankaj, "CTC – Problem Statement," 10 March 2021. [Online]. Available: <https://theailearner.com/2021/03/10/ctc-problem-statement>. [Accessed 26 June 2022].
- [67] A. Graves, S. Fernández, F. Gomez and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," ICML 2006 - Proceedings of the 23rd International Conference on Machine Learning, pp. 369-376, 2006.
- [68] A. Hannun, "Sequence Modeling with CTC," Distill.
- [69] P. B. C. Quoc, "Nhận Dạng Chữ Tiếng Việt - Vietnamese OCR," 31 October 2018. [Online]. Available: <https://pbcquoc.github.io/vietnamese-ocr>. [Accessed 26 June 2022].

- [70] TensorFlow Authors, "Neural machine translation with attention," 23 February 2022. [Online]. Available: https://www.tensorflow.org/text/tutorials/nmt_with_attention. [Accessed 27 June 2022].
- [71] M. D. Zeiler, "Adadelta: an adaptive learning rate method," arXiv preprint arXiv:1212.5701, 2012.
- [72] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," Soviet physics. Doklady, vol. 10, pp. 707-710, 1965.
- [73] C. Bartz, H. Yang và C. Meinel, "STN-OCR: A single Neural Network for Text Detection and Text Recognition," arXiv preprint arXiv:1707.08831, 2017.