**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA CÔNG NGHỆ THÔNG TIN 1**



**ĐỒ ÁN TỐT NGHIỆP**

**ĐỀ TÀI**

**NHẬN DIỆN CHỮ TIẾNG VIỆT VÀ ỨNG DỤNG VÀO THIẾT BỊ ĐỌC SÁCH CHO NGƯỜI KHIẾM THỊ TẠI THƯ VIỆN**

|  |  |  |
| --- | --- | --- |
| **Giảng viên hướng dẫn:** | **TS. NGUYỄN TRỌNG KHÁNH** | |
| **Sinh viên thực hiện:** | | **NGUYỄN VŨ CHIẾN** | |
| **Mã sinh viên:** | | **B17DCCN085** | |
| **Lớp:** | | **E17CN01** | |
| **Khoá:** | | **2017 - 2022** | |
| **Hệ:** | | **ĐẠI HỌC CHÍNH QUY** | |

**Hà Nội – 12/2021**

**Hà Nội, tháng 11 năm 2017**

***Hà Nội, 12/2015***

**C VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA CÔNG NGHỆ THÔNG TIN 1**



**ĐỒ ÁN TỐT NGHIỆP**

**ĐỀ TÀI**

**NGHIÊN CỨU PHƯƠNG PHÁP NHẬN DẠNG KÝ TỰ QUANG HỌC VÀ XÂY DỰNG HỆ THỐNG NHẬN DẠNG CHỨNG MINH THƯ NHÂN DÂN**

|  |  |  |
| --- | --- | --- |
| **Giảng viên hướng dẫn:** | **TS. NGUYỄN TRỌNG KHÁNH** | |
| **Sinh viên thực hiện:** | | **CHU VĂN AN** | |
| **Mã sinh viên:** | | **B16DCCN001** | |
| **Lớp:** | | **D16CNPM1** | |
| **Khoá:** | | **2016 - 2021** | |
| **Hệ:** | | **ĐẠI HỌC CHÍNH QUY** | |

**Hà Nội, tháng 11 năm 2017**

***Hà Nội, 12/2015***

**C VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA CÔNG NGHỆ THÔNG TIN 1**



**ĐỒ ÁN TỐT NGHIỆP**

**ĐỀ TÀI**

**NGHIÊN CỨU PHƯƠNG PHÁP NHẬN DẠNG KÍ TỰ QUANG HỌC VÀ XÂY DỰNG HỆ THỐNG NHẬN DẠNG CHỨNG MINH THƯ NHÂN DÂN**

|  |  |  |
| --- | --- | --- |
| **Giảng viên hướng dẫn:** | **TS. NGUYỄN TRỌNG KHÁNH** | |
| **Sinh viên thực hiện:** | | **CHU VĂN AN** | |
| **Mã sinh viên:** | | **B16DCCN001** | |
| **Lớp:** | | **D16CNPM1** | |
| **Khoá:** | | **2016 - 2021** | |
| **Hệ:** | | **ĐẠI HỌC CHÍNH QUY** | |

**Hà Nội, tháng 11 năm 2017**

***Hà Nội, 12/2015***

**C VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA CÔNG NGHỆ THÔNG TIN 1**



**ĐỒ ÁN TỐT NGHIỆP**

**ĐỀ TÀI**

**NGHIÊN CỨU PHƯƠNG PHÁP NHẬN DẠNG KÝ TỰ QUANG HỌC VÀ XÂY DỰNG HỆ THỐNG NHẬN DẠNG CHỨNG MINH THƯ NHÂN DÂN**

|  |  |  |
| --- | --- | --- |
| **Giảng viên hướng dẫn:** | **TS. NGUYỄN TRỌNG KHÁNH** | |
| **Sinh viên thực hiện:** | | **CHU VĂN AN** | |
| **Mã sinh viên:** | | **B16DCCN001** | |
| **Lớp:** | | **D16CNPM1** | |
| **Khoá:** | | **2016 - 2021** | |
| **Hệ:** | | **ĐẠI HỌC CHÍNH QUY** | |

**Hà Nội, tháng 11 năm 2017**

***Hà Nội, 12/2015***

# LỜI CẢM ƠN

---

Hà Nội, tháng 12 năm 2021

**Sinh viên**

**Nguyễn Vũ Chiến**

**NHẬN XÉT, ĐÁNH GIÁ, CHO ĐIỂM**

**(Của giảng viên hướng dẫn)**

**Điểm: (bằng chữ: )**

**Đồng ý/Không đồng ý** cho sinh viên bảo vệ trước hội đồng chấm tốt nghiệp.

Hà Nội, tháng 12 năm 2021

**CÁN BỘ - GIẢNG VIÊN HƯỚNG DẪN**

**NGUYỄN TRỌNG KHÁNH**

**NHẬN XÉT, ĐÁNH GIÁ, CHO ĐIỂM**

**(Của giảng viên phản biện)**

**Điểm: (bằng chữ: )**

**Đồng ý/Không đồng ý** cho sinh viên bảo vệ trước hội đồng chấm tốt nghiệp.

Hà Nội, tháng 12 năm 2021

**CÁN BỘ - GIẢNG VIÊN PHẢN BIỆN**

# MỤC LỤC

[LỜI CẢM ƠN 2](#_Toc88380928)

[MỤC LỤC 5](#_Toc88380929)

[DANH MỤC HÌNH ẢNH – chưa chỉnh sửa - 7](#_Toc88380930)

[DANH MỤC CÁC BẢNG – chưa chỉnh sửa - 9](#_Toc88380931)

[DANH MỤC TỪ VIẾT TẮT 10](#_Toc88380932)

[LỜI MỞ ĐẦU 11](#_Toc88380933)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ 15](#_Toc88380934)

[**1. Giới thiệu bài toán nhận dạng kí tự quang học** 15](#_Toc88380935)

[**2. Tổng quan các bước nhận dạng kí tự quang học** 17](#_Toc88380936)

[**3. Các phương pháp chuyển văn bản thành âm thanh** 17](#_Toc88380937)

[**4. Mục tiêu và phạm vi đồ án** 19](#_Toc88380938)

[II. NHẬN DẠNG VĂN BẢN TIẾNG VIỆT 20](#_Toc88380939)

[**1. Các phương pháp nhận dạng trích xuất thông tin từ ảnh** 20](#_Toc88380940)

[**1.1. Các phương pháp tiền xử lý hình ảnh** 20](#_Toc88380941)

[**1.2. Các phương pháp phát hiện vùng kí tự** 24](#_Toc88380942)

[**1.3. Các phương pháp nhận dạng chuỗi kí tự** 29](#_Toc88380943)

[**2. Xây dựng mô hình** 46](#_Toc88380944)

[**3. Kết luận** 47](#_Toc88380945)

[----ĐANG HOÀN THIỆN---- 48](#_Toc88380946)

[CHƯƠNG 3: THỬ NGHIỆM VÀ ĐÁNH GIÁ 48](#_Toc88380947)

[**1. Kịch bản hệ thống** 48](#_Toc88380948)

[**2. Xây dựng hệ thống** 48](#_Toc88380949)

[**2.1. Phân tích hệ thống** 48](#_Toc88380950)

[**2.2. Thiết kế hệ thống** 49](#_Toc88380951)

[**3. Huấn luyện mô hình và nhận dạng trích xuất thông tin** 49](#_Toc88380952)

[**3.1. Huấn luyện mô hình** 50](#_Toc88380953)

[**3.2. Nhận dạng trích xuất thông tin** 51](#_Toc88380954)

[**4. Thử nghiệm hệ thống** 52](#_Toc88380955)

[**4.1. Cài đặt huấn luyện các mô hình** 52](#_Toc88380956)

[**4.2. Nhận dạng trích xuất thông tin** 57](#_Toc88380957)

[**5. Kết luận** 62](#_Toc88380958)

[CHƯƠNG 4: KẾT LUẬN 64](#_Toc88380959)

[**1. Đóng góp của đồ án** 64](#_Toc88380960)

[**2. Hướng phát triển tiếp theo** 64](#_Toc88380961)

[TÀI LIỆU THAM KHẢO 65](#_Toc88380962)

# DANH MỤC HÌNH ẢNH – chưa chỉnh sửa -

[Hình 1.1: Công nghệ OCR giúp đưa tài liệu lên hệ thống và trích xuất thông tin nhanh chóng [46] 14](#_Toc61207341)

[Hình 1.2: Xác định xe vi phạm thông qua nhận diện biển số xe [47] 14](#_Toc61207342)

[Hình 2.1: Một số bài toán trong lĩnh vực thị giác máy tính [65] 17](#_Toc61207343)

[Hình 2.2: Kết quả của bài toán phân đoạn ngữ nghĩa [45] 21](#_Toc61207349)

[Hình 2.3: Kết quả của bài toán Phân đoạn cá thể [45] 21](#_Toc61207350)

[Hình 2.4: Kiến trúc mô hình U-Net [1] 22](#_Toc61207351)

[Hình 2.5: Minh họa các bước chính trong giải thuật SIFT [51] 25](#_Toc61207352)

[Hình 2.6: Kiến trúc mạng VGG16 [48] 26](#_Toc61207353)

[Hình 2.7: Cấu trúc các mạng VGG [5] 27](#_Toc61207354)

[Hình 2.8: Mô phỏng kiến trúc mô hình CRAFT [12] 30](#_Toc61207355)

[Hình 2.9: Kiến trúc mạng VGG16BN 30](#_Toc61207356)

[Hình 2.10: Mạng VGG16BN áp dụng trong CRAFT 31](#_Toc61207357)

[Hình 2.11: Kiến trúc CRNN [18] 32](#_Toc61207358)

[Hình 2.12: Trường tiếp nhận. Mỗi vectơ trong chuỗi đặc trưng được trích xuất được liên kết với một trường tiếp nhận trên hình ảnh đầu vào và có thể được coi là vectơ đặc trưng của trường đó [18]. 34](#_Toc61207359)

[Hình 2.13: Cấu trúc bên trong tế bào LSTM 35](#_Toc61207360)

[Hình 2.14. Cấu trúc của LSTM hai chiều sâu. Kết hợp LSTM xuôi (trái sang phải) và ngược (phải sang trái) sẽ tạo ra LSTM hai chiều. Xếp chồng nhiều LSTM hai chiều dẫn đến một LSTM hai chiều sâu. 37](#_Toc61207361)

[Hình 2.15: So sánh hiệu quả của các mô hình STR hiện có và thực nghiệm trên các mô hình đó của nhóm tác giả [28] 39](#_Toc61207362)

[Hình 2.16: Kiến trúc của STN. Mạng cục bộ hóa định vị các điểm cơ sở C, bộ tạo lưới tạo ra lưới lấy mẫu P, bộ lấy mẫu V tạo hình ảnh chuẩn hóa I' từ hình ảnh đầu vào I và P [25] 41](#_Toc61207363)

[Hình 2.17: Tập điểm cơ sở và bộ chuyển đổi TPS [25] 41](#_Toc61207364)

[Hình 2.18: STN căn chỉnh một số hình ảnh chuỗi kí tự thường gặp [25] 43](#_Toc61207365)

[Hình 2.19: Kiến trúc của mạng cục bộ trong STN [28] 44](#_Toc61207366)

[Hình 2.20: Sự khác biệt giữa một khối thông thường (trái) và một khối phần dư (phải) [44] 45](#_Toc61207367)

[Hình 2.21: Mạng ResNet được sử dụng trong TBRA [28] 46](#_Toc61207368)

[Hình 2.22: Cơ chế seq2seq phổ biến (không có attention) 47](#_Toc61207369)

[Hình 2.23: Cơ chế seq2seq có attention 47](#_Toc61207370)

[Hình 2.24: Mô hình nhận dạng trích xuất thông tin chứng minh thư nhân dân 49](#_Toc61207372)

[Hình 2.25: Kiến trúc mạng phân loại của mô hình 50](#_Toc61207373)

[Hình 2.26: Kết quả qua các bước tiền xử lý và phát hiện vùng kí tự 51](#_Toc61207374)

[Hình 3.1: Biểu đồ ca sử dụng trong hệ thống 53](#_Toc61207375)

[Hình 3.2: Kiến trúc chung của hệ thống 54](#_Toc61207376)

[Hình 3.3. Quy trình chung huấn luyện và nhận dạng trích xuất thông tin 55](#_Toc61207377)

[Hình 3.4: Quy trình huấn luyện mô hình 56](#_Toc61207378)

[Hình 3.5: Quy trình nhận dạng trích xuất thông tin chứng minh thư nhân dân của hệ thống 57](#_Toc61207379)

[Hình 3.6: Thuật toán hậu xử lý xác định các chuỗi kí tự thuộc trường nào 58](#_Toc61207380)

[Hình 3.7: Thuật toán hậu xử lý kết quả trường họ tên 59](#_Toc61207381)

[Hình 3.8: Thuật toán hậu xử lý kết quả trường địa chỉ 60](#_Toc61207382)

[Hình 3.9: Hình ảnh chứng minh nhân dân và nhãn tương ứng 62](#_Toc61207383)

[Hình 3.10: Kết quả huấn luyện mô hình phân đoạn 63](#_Toc61207384)

[Hình 3.11: Kết quả huấn luyện mô hình phân loại 64](#_Toc61207385)

[Hình 3.12: Công cụ gán nhãn kí tự 65](#_Toc61207386)

[Hình 3.13: Tham số đào tạo TBRA 66](#_Toc61207387)

[Hình 3.14: Kết quả huấn luyện TBRA 67](#_Toc61207388)

[Hình 3.15: Trường hợp phát hiện vùng kí tự sai 71](#_Toc61207389)

[Hình 3.16: Nhận dạng trích xuất thông tin đúng 72](#_Toc61207390)

[Hình 3.17: Nhận dạng trích xuất thông tin sai do phát hiện vùng kí tự sai 72](#_Toc61207391)

[Hình 3.18: Không phân đoạn được ảnh 72](#_Toc61207392)

# DANH MỤC CÁC BẢNG – chưa chỉnh sửa -

[Bảng 3.1: Môi trường cài đặt thử nghiệm việc huấn luyện các mô hình 61](#_Toc60859292)

[Bảng 3.2: Cấu hình hệ thống thử nghiệm 67](#_Toc60859293)

[Bảng 3.3: Các thư viện chính được sử dụng trong máy chủ hệ thống 68](#_Toc60859294)

[Bảng 3.4: Các công nghệ sử dụng trong máy chủ web và trình duyệt phía người dùng 68](#_Toc60859295)

[Bảng 3.5: Kết quả thử nghiệm hệ thống 71](#_Toc60859296)

# DANH MỤC TỪ VIẾT TẮT

|  |  |
| --- | --- |
| OCR: Optical Character Recognition | Nhận dạng kí tự quang học |
| AI: Artificial Intelligence | Trí tuệ nhân tạo |
| CV: Computer Vision | Thị giác máy tính |
| STR: Scene Text Recognition | Nhận dạng văn bản cảnh |
| CNN: Convolution Neural Network | Mạng nơ-ron tích chập |
| DCNN: Deep Convolutional Neural Cetwork | Mạng nơ-ron tích chập sâu |
| DL: Deep Learning | Học sâu |
| LSTM: Long Short Term Memory | Bộ nhớ dài – ngắn hạn |
| SW: Sliding Windows | Cửa sổ trượt |
| CC: Connected Components | Các thành phần kết nối |
| HOG: Histogram of oriented gradients | Biểu đồ độ dốc định hướng |
| STN: Spatial Transformer Networks | Mạng biến áp không gian |
|  |  |
|  |  |
|  |  |
|  |  |

# LỜI MỞ ĐẦU

Ngày nay, trí tuệ nhân tạo đang được ứng dụng rộng rãi và mang đến những giải pháp thiết thực trong các vấn để của đời sống xã hội, khiến cho cuộc sống của mỗi người trở nên dễ dàng, tiện lợi đặc biệt đối với người khuyết tật. Nhờ các ứng dụng trí tuệ nhận tạo mà những khó khăn, trở ngại đã dần được loại bỏ, giúp nâng cao chất lượng cuộc sống của họ. Trong đó, lĩnh vực thị giác máy tính đang được sử dụng như một giải pháp để thay thế con mắt của người khiếm thị, với các hệ thống trợ giúp nhận diện vật thế trong không gian để trợ giúp việc di chuyển hay các trợ lý ảo có thể trợ giúp sử dụng vật dụng trong nhà, hay số hoá các đầu sách dưới dạng sách nói – mà một trong những giải pháp đã và đang được phát triển cho việc số hoá là sử dụng phương pháp nhận dạng kí tự quang học (Optical Character Recognition - OCR), được hình thành từ lĩnh nghiên cứu về nhận dạng mẫu, trí tuệ nhân tạo và xử lý ảnh giúp trích xuất các văn bản trong hình ảnh thành văn bản số.

Với mong muốn tìm hiểu về phương pháp nhận dạng kí tự quang học cũng như đóng góp thêm vào kho ứng dụng về nhận dạng một hệ thống nhận dạng thiết thực, hữu ích. Đồ án thực hiện nghiên cứu các bước trong bài toán OCR và đề xuất giải pháp ứng dụng OCR trong nhận dạng văn bản tiếng Việt áp dụng trong các thư viện, giúp người khiếm thị có thể tiếp cận được nhiều đầu sách hơn bằng cách trực tiếp chuyển đổi sách giấy thành âm thanh.

Nội dung đồ án bao gồm 4 chương:

* Chương 1: Giới thiệu chung
* Chương 2: Cơ sở lý thuyết và công nghệ
* Chương 3: Đề xuất phương pháp
* Chương 4: Thử nghiệm và đanh giá

**CHƯƠNG 1: GIỚI THIỆU CHUNG**

**1. Ứng dụng học sâu trong hỗ trợ người khiếm thị tiếp cận sách**

Trong những năm qua, để phục vụ người khiếm thị, các thư viện công cộng đã xây dựng, tổ chức nhiều phòng đọc với các loại hình tài liệu như sách chữ nổi Braille, sách nói, sách nói kỹ thuật số, sách minh họa nổi, tài liệu đồ họa nổi, máy tính cùng các phần mềm chuyên dụng... Các phương tiện, thiết bị ứng dụng công nghệ thông tin phong phú, đa dạng giúp người khiếm thị tiếp cận việc tìm và đọc sách dễ dàng hơn. Không chỉ đọc sách chữ nổi, tạo ra sách nói, các dịch vụ hướng tới người khiếm thị đã tạo ra nhiều hoạt động hướng dẫn, giúp đỡ người khiếm thị có cơ hội tiếp cận thông tin, tri thức, tạo ra sản phẩm, giúp người khiếm thị tiếp cận và tham gia cuộc sống hằng ngày. Cùng với đó, các giải pháp về công nghệ học sâu và trí tuệ nhân tạo cũng được ứng dụng mạnh mẽ. Trong đó có các chương trình phần mềm cho phép người dùng khiếm thị đọc văn bản được hiển thị trên màn hình máy tính hoặc sách báo với bộ tổng hợp giọng nói hoặc màn hình chữ nổi. Người dùng gửi lệnh bằng cách nhấn các tổ hợp phím khác nhau trên bàn phím máy tính hoặc màn hình chữ nổi để hướng dẫn bộ tổng hợp giọng nói phải nói gì và âm thanh sẽ tự động phát khi có sự thay đổi nội dung trên màn hình. Một lệnh có thể hướng dẫn bộ tổng hợp đọc hoặc đánh vần một từ, đọc một dòng hoặc toàn màn hình văn bản, tìm một chuỗi văn bản trên màn hình. Ngoài ra, nó cho phép người dùng thực hiện các chức năng nâng cao hơn, chẳng hạn như định vị văn bản được hiển thị bằng một màu nhất định, đọc các phần được chỉ định trước của màn hình theo yêu cầu, đọc văn bản được đánh dấu và xác định lựa chọn đang hoạt động trong menu. Người dùng cũng có thể sử dụng trình kiểm tra chính tả trong trình xử lý văn bản hoặc đọc các ô của bảng tính bằng trình đọc màn hình.

**2. Mục tiêu và phạm vi nghiên cứu**

Đồ án nghiên cứu và đưa ra giải pháp cho bài toán nhận diện và đọc sách giấy cho người khiếm thị. Nội dung nghiên cứu tâp trung vào nghiên cứu các giải pháp nhận diện ký tự từ ảnh quét được bằng thiết bị chuyên dụng và chuyển văn bản dự đoán thành giọng nói. Đầu vào của bài toán được giả định là hình chụp của trang sách với điều kiện ảnh sáng đủ tốt và nền không quá phức tạp, mục tiêu hướng đến sau khi xây dựng hệ thống là có thể đạt hiệu năng cao trong nhận diện và chuyển đổi văn bản nhận diện thành âm thanh và thông qua API để phát thành giọng đọc.

**3. Một số nghiên cứu liên quan**

**\* Kurzweil 1000**

Phần mềm đọc và quét từng đoạt giải thưởng giúp những người khiếm thị có thể truy cập văn bản điện tử hoặc bản in. Nó kết hợp các công nghệ máy đọc truyền thống như quét, xử lý hình ảnh và chuyển văn bản thành giọng nói với các công cụ giao tiếp và năng suất để dễ dàng và nâng cao trải nghiệm đọc, viết và học tập của người dùng. Phần mềm nói to văn bản bằng nhiều giọng đọc tự nhiên có thể được sửa đổi theo sở thích cá nhân.

Ngoài ra, phần mềm cung cấp cho người dùng khả năng viết và chỉnh sửa tài liệu, hoàn thành các biểu mẫu đơn giản một cách độc lập. Nó bao gồm các tính năng ghi chú, tóm tắt nội dung và lập dàn ý cho văn bản.



*Hình 1.1. Hệ thống Kurzweil 1000 với máy quét hình ảnh*

**\* OrCam MyEye**

OrCam MyEye là thiết bị chuyển văn bản thành giọng nói tiên tiến nhất hiện nay. Thiết bị thông minh khai thác sức mạnh của Thị giác nhân tạo (Artificial Vision) để hỗ trợ những người khiếm thị. Nó có thể đọc văn bản ngay lập tức từ bất kỳ bề mặt in hoặc kỹ thuật số nào. Một trong những thuộc tính hữu ích nhất của thiết bị là thiết bị không cần kết nối Internet để hoạt động.

Có hai cách mà người dùng có thể sử dụng thiết bị để đọc văn bản. Đầu tiên là tính năng đọc văn bản đơn giản. Tính năng này sẽ đọc văn bản từ đầu trang cho đến cuối trang. Tùy chọn thứ hai là sử dụng tính năng đọc văn bản thông minh. Tính năng này cho phép người dùng hướng dẫn bằng lời nói thiết bị để tìm thông tin cụ thể trong văn bản. Ví dụ, khi đọc hóa đơn tiền điện, bạn có thể hướng dẫn thiết bị tìm "tổng số tiền đến hạn". Video này trình bày cách hoạt động của tính năng đọc thông minh.



*Hình 1.2. Thiết bị OrCam MyEye*

**4. Kết luận**

Chương 1 đã giới thiệu về bài toán nhận dạng kí tự quang học cũng như đề xuất mục tiêu xây dựng hệ thống nhận dạng ký tự tiếng Việt và những sản phẩm với nghiên cứu tương tự trên thị trường.

Trong chương tiếp theo, đồ án sẽ trình bày chi tiết hơn về cơ sở lý thuyết và các nghiên cứu hiện có cho từng bước trong bài toán nhận dạng và chuyển đổi văn bản tiếng Việt thành giọng nói.

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ

Chương này giới thiệu tổng quan về bài toán nhận dạng kí tự quang học cũng như các bước thực hiện của một bài toán nhận dạng kí tự quang học điển hình, đồng thời giới thiệu về bài toán nhận dạng văn bản tiếng Việt, một bài toán con trong nhận dạng kí tự quang học.

**I. NỀN TÀNG CHUNG VỀ CÔNG NGHỆ**

## **1. Giới thiệu bài toán nhận dạng kí tự quang học**

Nhận dạng kí tự quang học là phương pháp giúp chuyển các hình ảnh của chữ viết tay hoặc chữ đánh máy thành các văn bản tài liệu. Nó là một trong những đề tài nghiên cứu hấp dẫn, có nhiều ứng dụng thực tiễn trong cuộc sống giảm thiểu thời gian công sức cho con người trong việc chuyển đổi ảnh scan, ảnh chụp tài liệu … thành dạng văn bản.

Hiện nay, các phần mềm có tích hợp ứng dụng hoặc phần mềm nhận diện ký tự đang được sử dụng phổ biến trong nhiều ngành nghề, đem lại hiệu quả công việc cao, tiết kiệm thời gian làm việc và giảm bớt nhân sự. Có thể kể đến trong đó là hành chính công và giao thông với nhiều ứng phổ biến.

\* Trong lĩnh vực hành chính công

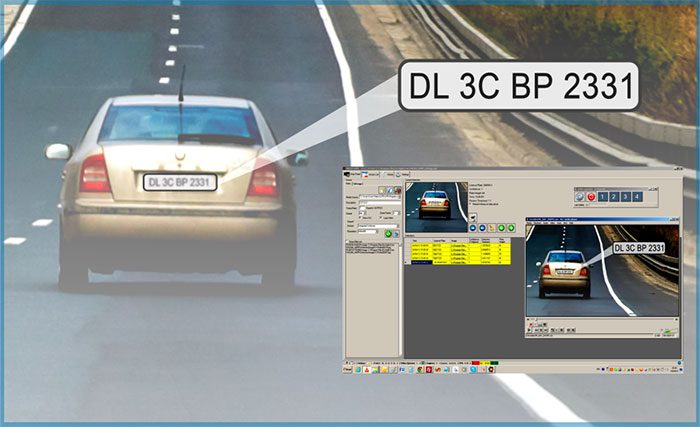
Công nghệ OCR đã mang lại rất nhiều tiện ích và rút ngắn thời gian xử lý các thủ tục hành chính. Bởi để phục vụ cho doanh nghiệp và người dân, các cơ quan cần tổng hợp và kiểm chứng nhiều danh mục thông tin khác nhau, cũng như bổ sung vào nguồn dữ liệu lớn để dần rút ngắn những thủ tục không cần thiết mỗi khi cần xử lý các yêu cầu pháp lý hướng đến cơ chế “một cửa”. Với sự kết với AI, OCR đã có thể nhận diện, trích xuất được các thông tin mong muốn từ những hồ sơ, văn bản được nộp vào hệ thống, từ đó giúp cho các thủ tục được thông suốt, đi đến đúng đơn vị cần xử lý. Đối với văn bản đánh máy, tỷ lệ chính xác của OCR lên đến hơn 98%. Tỷ lệ chính xác là rất cao tuy nhiên tỷ lệ nhỏ sai sót cũng có thể gây ra hậu quả lớn. Do đó, các cơ quan hành chính sử dụng công nghệ OCR như một phương tiện để hỗ trợ phân loại và xử lý văn bản đầu vào. Vậy nên, việc áp dụng công nghệ OCR vào hệ thống giúp quản lý được công việc hiệu quả, chuẩn hóa hoạt động của từng bộ phận, đơn giản hoá các quy trình, giảm gánh nặng, sai sót khi kiểm tra thủ tục hành chính, tăng năng suất lao động và tối ưu hoá chi phí cho doanh nghiệp, người dân.



Hình 2.1: Công nghệ OCR giúp đưa tài liệu lên hệ thống và trích xuất thông tin nhanh chóng [46]

\*Trong lĩnh vực giao thông

Ở các thành phố lớn hoặc trên các tuyến cao tốc, lưu lượng phương tiện giao thông ngày càng tăng tuy nhiên hệ thống giao thông hạ tầng lại chưa đáp ứng đủ gây ra tình trạng vi phạm luật lệ giao thông như chạy quá tốc độ, lấn làn, dừng đỗ sai quy định. Công nghệ OCR đã được áp dụng để hỗ trợ lực lượng chức năng có thể quản lý giao thông, xử phạt những hành vi vi phạm. Thông qua hệ thống camera giám sát, phần mềm nhận dạng kí tự quang học nhận diện biển số các phương tiện có hành vi vi phạm giao thông như vượt đèn đỏ, sai làn đường, không đội mũ bảo hiểm…Từ những hình ảnh này sau đó lấy thông tin của chủ phương tiện rồi tiến hành xử lý vi phạm. Ngoài ra công nghệ này còn hỗ trợ lưu trữ các hình ảnh, tìm hiểu diễn biến, nguyên nhân các vụ tai nạn.



Hình 2.2: Xác định xe vi phạm thông qua nhận diện biển số xe [47]

## **2. Tổng quan các bước nhận dạng kí tự quang học**

Một hệ thống nhận dạng kí tự quang học điển hình bao gồm các bước:

\* Thu thập hình ảnh: là quá trình chuyển hình ảnh thực của các đối tượng về dạng ảnh số, video số bằng cách sử dụng các thiết bị quay, chụp như máy ảnh số (camera), webcame hoặc các thiết bị smartphone. Đây thao tác cơ bản đầu tiên để cung cấp dữ liệu ảnh đầu vào cho các ứng dụng xử lý ảnh, xử lý video.

\* Tiền xử lý hình ảnh: là quá trình áp dụng các kĩ thuật xử lý nhằm phát hiện được đối tượng quan tâm trong ảnh và đồng thời xác định loại văn bản cần đọc để phục vụ cho công đoạn tiếp theo.

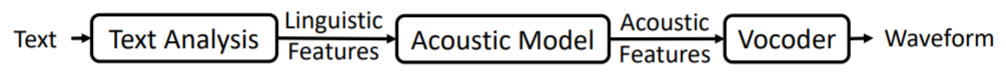
\* Phát hiện các vùng chứa kí tự trong ảnh: xác định các vùng chứa kí tự trong ảnh, các vùng có thể ở mức kí tự, từ hoặc dòng bằng các kĩ thuật xử lý ảnh cổ điển hoặc học máy.

\* Nhận dạng kí tự: là công đoạn xác định nhãn kí tự, từ hay dòng của hình ảnh được cắt từ bước phát hiện. Quá trình nhận dạng hiện nay thường áp dụng các mô hình học máy tổng hợp gồm một mô hình trích xuất các đặc trưng của ảnh và mô hình hồi quy để giúp nắm bắt thông tin ngữ cảnh của chuỗi kí tự.

## **3. Các phương pháp chuyển văn bản thành âm thanh**

https://arxiv.org/pdf/2106.15561.pdf

Hiện nay, bài toán chuyển đổi văn bản thành giọng nói – Text to speech (TTS) hoặc tổng hợp giọng nói, nhằm mục đích tổng hợp văn bản đã cho thành giọng nói dễ hiểu và tự nhiên, là một chủ đề nghiên cứu nóng trong cộng đồng lời nói, ngôn ngữ và máy học và có ứng dụng rộng rãi trong ngành. Cũng như sự phát triển của học sâu (deep learning) và trí tuệ nhân tạo, TTS dựa trên mạng neuron đã cải thiện đáng kể chất lượng kết quả đầu ra của bài toán.



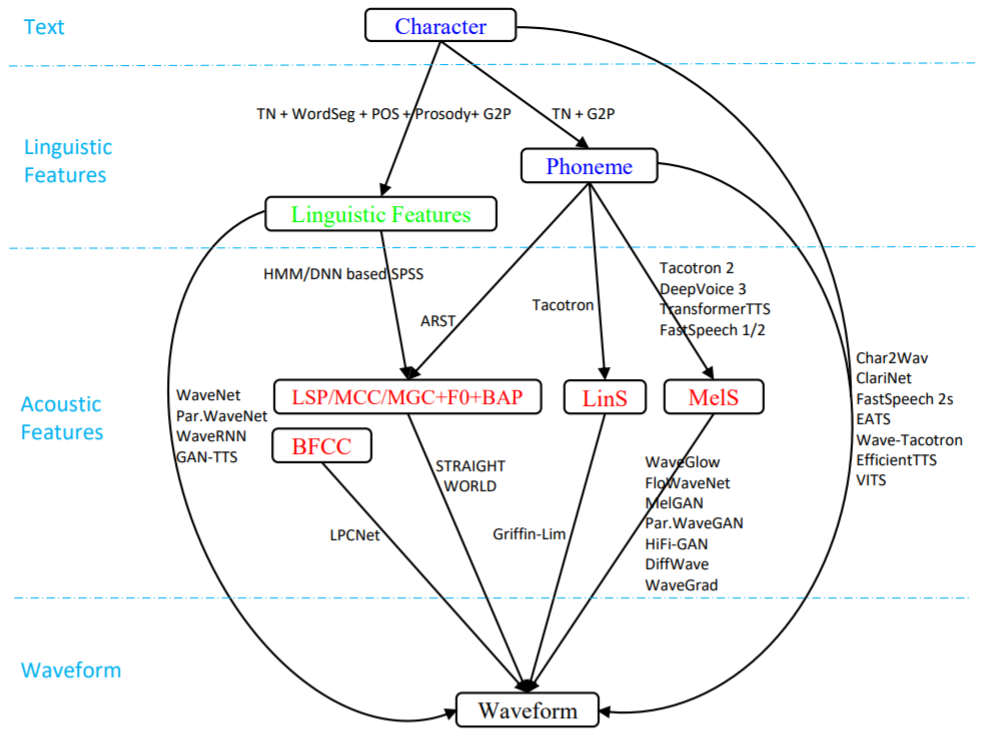
Hình 2.3: Mô hình các thành phần cơ bản của một hệ thống TTS

Một hệ thống TTS hiện đại bao gồm ba thành phần cơ bản:

- Module phân tích văn bản (Text Analysis): chuyển đổi một chuỗi văn bản thành các đặc điểm ngôn ngữ.

- Mô hình âm thanh (Acoustic Model): tạo ra các đặc điểm âm thanh từ các đặc điểm ngôn ngữ.

- Bộ mã hóa (Vocoder): tổng hợp ra các dạng song âm thanh. từ các đặc điểm âm học.



Hình 2.4: Mô hình luồng dữ liệu từ văn bản thành giọng nói

Cách hoạt động của TTS theo luồng dữ liệu từ văn bản sang dạng song được tóm tắt như trong hình 3.2. Có một số biểu diễn dữ liệu trong quá trình chuyển đổi văn bản thành giọng nói:

1) Các ký tự (Text), là định dạng thô của văn bản.

2) Các đặc điểm của ngôn ngữ (Linguistic Feature), có được thông qua phân tích văn bản và chứa thông tin ngữ cảnh phong phú về cách phát âm và ngữ điệu. Ngoài ra có âm vị là một trong những yếu tố quan trọng nhất trong các đặc điểm ngôn ngữ và thường được sử dụng để biểu thị văn bản trong các mô hình TTS dựa trên mạng neuron.

3) Các tính năng âm học (Acoustic Feature), là các đại diện trừu tượng của dạng sóng giọng nói. Theo những thống kê tổng hợp về tham số đặc trưng cho giọng nói, cặp quang phổ vạch - LSP (line spectral pairs) [135], hệ số mel-cepstral - MCC (mel-cepstral coefficients) [82], hệ số tổng quát mel MGC - (mel-generalized coefficients) [355], F0 và các đặc điểm của dải BAP - (band aperiodicities) [156, 157] được sử dụng như các đặc trưng âm học, có thể dễ dàng chuyển đổi thành dạng sóng thông qua các bộ định tuyến như STRAIGHT [155] và WORLD [238]. Trong các mô hình TTS end- to-end dựa trên mạng neuron, phổ Mel (mel-spectrograms) hoặc phổ tuyến tính (linear-spectrograms) thường được sử dụng làm các đặc trưng âm thanh, được chuyển đổi thành dạng sóng bằng cách sử dụng các bộ mã hóa dựa trên mạng neuron.

4) Dạng sóng, định dạng cuối cùng của giọng nói. Có thể có các luồng dữ liệu khác nhau từ văn bản sang dạng sóng, bao gồm:

1) Ký tự → đặc điểm ngôn ngữ → đặc điểm âm thanh → dạng sóng;

2) Ký tự → âm vị → đặc điểm âm học → dạng sóng;

3) Ký tự → đặc điểm ngôn ngữ → dạng sóng;

4) Ký tự → âm vị → đặc điểm âm học → dạng sóng;

5) Ký tự → âm vị → dạng sóng, hoặc ký tự → dạng sóng.

## **4. Mục tiêu và phạm vi đồ án**

Mục tiêu chính của đồ án là nghiên cứu phương pháp nhận dạng kí tự quang học nói chung và áp dụng trong việc nhận dạng văn bản tiếng Việt cũng như xây dựng hệ thống chuyển đổi văn bản thành giọng nói hỗ trợ cho người khiếm thị trong các thư viện. Cụ thể đồ án sẽ:

- Về mặt lý thuyết, tìm hiểu về các phương pháp đang được sử dụng trong các bước của nhận dạng ký tự tiếng Việt bao gồm các mô hình phát hiện kí tự, mô hình nhận dạng kí tự và các kĩ thuật hậu xử lý.

- Về mặt ứng dụng, đồ án đặt mục tiêu xây dựng hệ thống có khả năng nhận dạng văn bản tiếng Việt từ trang sách sau đó thông tin được chuyển đổi thành giọng nói sử dụng API của Google.

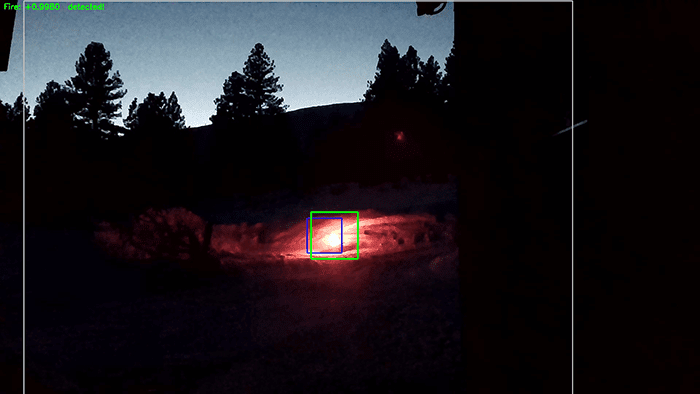
# II. NHẬN DẠNG VĂN BẢN TIẾNG VIỆT

## **1. Các phương pháp nhận dạng trích xuất thông tin từ ảnh**

Phần này sẽ giới thiệu về các phương pháp tiền xử lý, phát hiện vùng ký tự và nhận dạng ký tự hiện có và được áp dụng rộng rãi trong các bài toán nhận diện ký tự.

### **1.1. Các phương pháp tiền xử lý hình ảnh**

Trước khi đi sâu vào từng phương pháp, ta cùng tìm hiểu về các nhóm bài toán trong thị giác máy tính (Computer Vision - CV). Bao gồm: phân lớp ảnh (Image classification), phân lớp và xác định vị trí (Classification with Localization), phát hiện đối tượng (Object detection), phân đoạn ảnh (Image segmentation).



Hình 2.5: Phát hiện một đám cháy nhỏ ở khoảng 30 mét [45]

https://www.pyimagesearch.com/2021/04/17/image-classification-basics/

**\* Phân lớp ảnh (Image classification):**

Mục tiêu của bài toán là phân tích hình ảnh đầu vào và trả về một nhãn phân loại hình ảnh. Nhãn luôn nằm trong một tập hợp các danh mục có thể được xác định trước. Hệ thống phân loại cũng có thể gán nhiều nhãn cho hình ảnh thông qua xác suất, chẳng hạn như {chó: 95%; mèo: 4%; gấu trúc: 1%}.

Kết quả bài toán phân lớp ảnh có thể áp dụng vào rất nhiều lĩnh vực như phân loại động vật, phân loại biển báo giao thông… Phân lớp ảnh cũng được cho là bài toán cơ sở cho một số bài toán khác trong CV. Trong thực tế bài toán này có khá nhiều thách thức:



Hình 2.6: Những yếu tố anht hưởng đến cách một hình ảnh của đối tượng được ghi lại [45]

* Đa dạng về góc nhìn (Viewpoint Variation): Đối tượng có thể được định hướng / xoay nhiều chiều phụ thuộc vào cách đối tượng được chụp.
* Đa dạng về tỉ lệ/ kích thước (Scale Variation): Với một chiêc áo có thể có nhiều size như S, M, L... Về mặt kỹ thuật, chúng đều giống nhau - một chiếc áo, nhưng chúng có kích thước khác nhau. Hơn nữa, cùng một chiếc áo size L sẽ trông khác biệt đáng kể khi nó được chụp gần so với khi nó được chụp từ xa hơn.
* Biến dạng (Deformation): Sự đa dạng hình ảnh của cùng một đối tượng có thể bị biến đổi theo các điều kiện khác nhau.
* Bị che khuất (Occlusion Variation): Đối tượng cần phân loại bị che khuất một phần.
* Điều kiện chiếu sáng (Illumination Variation): Cùng một ảnh chụp một vật nhưng ở những thiết lập chiếu sáng khác nhau có thể có những khác biệt đáng kể, gây sai sót trong quá trình phân loại.
* Ảnh hưởng bởi bối cảnh (Background Clutter): Những hình ảnh có "nhiễu" hoặc có rất nhiều đối tượng trong đó. Việc tìm kiếm một đối tượng cụ thể trong ảnh cũng gây khó khăn đối với con người. Nên sẽ khó khăn hơn nữa với máy tính không hiểu ngữ nghĩa của hình ảnh. Ngoài ra còn trường hợp đa dạng về biến thể trong một nhãn: Nhãn cần phân loại có rất nhiều loại ví dụ mèo thì có mèo tam thể, mèo mướp.

**\* Phân lớp và định vị (Classification with Localization)**

https://datalya.com/blog/machine-learning/image-classification-with-localization

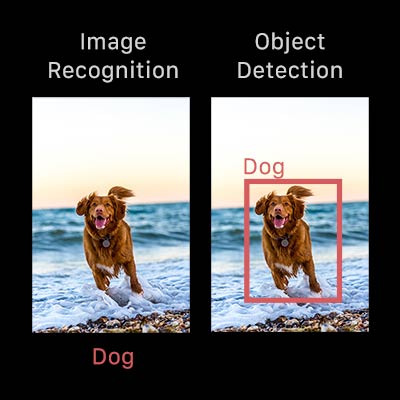
Trong bài toán phân lớp và định vị, một thuật toán được đào tạo có giám sát (supervised algorithm) để dự đoán lớp cũng như hộp giới hạn xung quanh đối tượng (bounding box) trong ảnh. Thuật ngữ ‘localization’ đề cập đến vị trí của đối tượng trong hình ảnh.

**\* Phát hiện đối tượng (Object detection)**

https://www.fritz.ai/object-detection/

Phát hiện đối tượng là một kỹ thuật thị giác máy tính để xác định và định vị các đối tượng trong một hình ảnh hoặc video. Cụ thể, phát hiện đối tượng vẽ các hộp giới hạn (bounding box) xung quanh các đối tượng được phát hiện này, cho phép xác định vị trí các đối tượng đang ở (hoặc cách chúng di chuyển qua, đối với đầu vào video) trong một cảnh nhất định.

Phát hiện đối tượng thường bị nhầm lẫn với nhận dạng hình ảnh. Nhận dạng hình ảnh gán nhãn cho hình ảnh. Hình ảnh một con chó được gắn nhãn "dog". Một bức ảnh của hai con chó, vẫn nhận được nhãn "dog". Mặt khác, phát hiện đối tượng, sẽ có một hộp xung quanh mỗi con chó và dán nhãn cho hộp đó là “dog”. Mô hình dự đoán vị trí của từng đối tượng và nhãn nào nên được áp dụng. Theo cách đó, phát hiện đối tượng cung cấp nhiều thông tin về hình ảnh hơn là nhận dạng.



Hình 2.7: Sự khác nhau giữa nhận diện hình ảnh và phá hiện đối tượng [45]

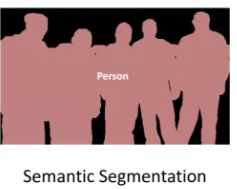
**\* Phân đoạn ảnh (Image segmentation).**

**https://phamdinhkhanh.github.io/2020/06/10/ImageSegmention.html#12-bài-toán-image-segmentation**

Bài toán sẽ phân chia một hình ảnh thành nhiều vùng ảnh khác nhau. Image Segmentation cũng có mục tiêu giống như object detection là phát hiện ra vùng ảnh chứa vật thể và gán nhãn phù hợp cho chúng. Tuy nhiên tiêu chuẩn về độ chính xác của Image Segmentation ở mức cao hơn so với Object Detection khi nó yêu cầu nhãn dự báo đúng tới từng pixel. Input của bài toán là một bức ảnh và output là một ma trận mask mà giá trị của từng pixel đã được gãn nhãn trên đó.

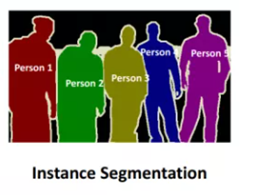
Bài toán phân đoạn hình ảnh có thể được chia làm 2 loại:

* Phân đoạn ngữ nghĩa (Sematic Segmentation): phân đoạn các vùng ảnh theo những nhãn khác nhau mà không phân biệt sự khác nhau giữa các đối tượng trong từng nhãn. Ví dụ trong bức ảnh xuất hiện 5 người, mức độ phân chia sẽ không xác định từng pixel thuộc về người nào.



Hình 2.8: Kết quả của bài toán phân đoạn ngữ nghĩa [64]

* Phân đoạn cá thể (Insane Segmentation): phân đoạn các vùng ảnh chi tiết đến từng đối tượng trong mỗi nhãn. Ví dụ: một ảnh có 5 người sẽ được phân chia segment chi tiết tới từng người 1, 2, … , 5.



Hình 2.9: Kết quả của bài toán Phân đoạn cá thể [64]

### **1.2. Các phương pháp phát hiện vùng kí tự**

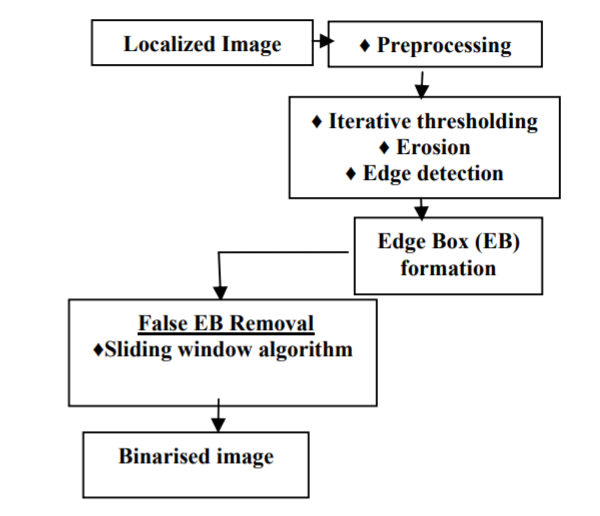
Bài toán phát hiện và nhận dạng văn bản đã có những phương pháp tối ưu hoá và đưa ra kết quả rất tốt trong những năm gần đây. Trong phần này, các phương pháp truyền thống và hiện đại với việc sử dụng học sâu trong việc phát hiện các vùng kí tự sẽ được giới thiệu cùng những ưu điểm, nhược điểm của chúng.

#### **1.2.1. Phương pháp truyền thống**

https://arxiv.org/ftp/arxiv/papers/1003/1003.3654.pdf

Trình phát hiện văn bản truyền thống tập trung vào việc phát triển các đặc trưng cấp thấp được làm thủ công để phân biệt các thành phần văn bản và không phải văn bản trong hình ảnh, một trong những phương pháp phổ biễn là cửa sổ trượt (Sliding Windows - SW).

Việc nhận diện và phân loại các pixel trên ảnh là thuộc ký tự hay nền cùng với mã hoá nhị phân ảnh là bước tiền xử lý quan trọng góp phần tăng tỉ lệ chính xác của các thuật toán OCR. Các kỹ thuật hiện tại đang áp dụng bao gồm: Mã hoá nhị phân toàn cục và Mã hoá nhị phân cục bộ. Trong phương pháp mã hóa nhị phân toàn cục, ngưỡng toàn cục được sử dụng cho tất cả các pixel trong ảnh và không phù hợp với ảnh tài liệu phức tạp và đã xuống cấp. Phương thức này tỏ ra nhanh chóng và mạnh mẽ cho các đoạn văn bản nhỏ. Mặt khác, các phương pháp mã hóa nhị phân cục bộ thay đổi ngưỡng sao cho thích ứng với từng vùng trên hình ảnh. Phương pháp mã hóa nhị phân cục bộ được đề xuất để khắc phục các nhược điểm gặp phải của phương pháp toàn cục. Phương pháp mã hóa nhị phân cục bộ có thể được cải thiện bằng cách tính toán các ngưỡng cục bộ trong các cửa sổ hoặc khu vực riêng biệt. Trong hầu hết các phương pháp này, kích thước và hình dạng của cửa sổ là các tham số được xác định trước. Quá trình mã hóa nhị phân có thể thu lại kết quả kém khi ranh giới của cửa sổ vượt qua các ký tự và có thể làm phát sinh các ký tự bị hỏng và những khoảng trống (dấu cách), điều này có thể gây ra các lỗi không mong muốn trong quá trình OCR hình ảnh nhị phân.



Hình 2.10: Các bước của kỹ thuật nhận diện bằng cửa sổ trượt

Đế áp dụng thuật toán cửa số trượt vào ảnh, cần các bước tiền xử lý:

- Tăng cường độ tương phản có điều kiện (Conditional Contrast Enhancement): Hình ảnh được chụp trong điều kiện ánh sáng kém có thể dẫn đến entropy thấp, điều này cần tăng độ tương phản của hình ảnh để xử lý tốt hơn. Do đó, entropy có thể được sử dụng như một chỉ báo nếu cần tăng độ tương phản của hình ảnh. Nếu entropy được tính toán của hình ảnh quá thấp, độ tương phản có thể được tăng lên, nếu không, các cạnh được phát hiện của các ký tự có thể không tạo thành các hình dạng khép kín.

- Làm mịn ảnh (Smothing): giảm thiểu ảnh hưởng của những nhiễu đối với quá trình phát hiện cạnh và thực hiện mở rộng thang màu xám cho hình ảnh được điều chỉnh để tăng thêm độ tương phản.

- Tạo ảnh xám (Grayscale extension): Có thể tính toán các giá trị thang độ xám tối đa và tối thiểu của hình ảnh được làm mịn. Nếu max\_S và min\_S gần nhau, ví dụ: max\_S - min\_S <80, hình ảnh thang độ xám sẽ đơn điệu với độ tương phản thấp. Phần mở rộng thang độ xám sẽ tăng độ tương phản thông qua khai thác toàn dải max\_S - min\_S = 255

- Áp dụng ngưỡng lặp lại (Iterative thresholding) để tạo ảnh nhị phân và xác định cạnh (Edge detection): Các cạnh sẽ được phát hiện từ hình ảnh được xử lý từ các bước trên. Các thuật toán phát hiện cạnh thông thường thuộc về bộ lọc thông cao, không phù hợp với các ảnh nền phức tạp. Ngưỡng tổng thể cố định thường sẽ hoạt động tốt đối với hình ảnh có nền đồng nhất, nhưng không hoạt động đối với nền có kết cấu và độ tương phản của các đối tượng thay đổi trong ảnh. Cũng cần duy trì tính liên tục của cạnh và tránh chồng chéo cạnh. Một thuật toán ngưỡng lặp lại và thuật toán xói mòn hình thái học được sử dụng ở đây để phát hiện các cạnh. Sau đó, xói mòn hình thái học được áp dụng để đảm bảo rằng các cạnh được phát hiện chỉ rộng một. Trong khi đó, điều này tránh được sự chồng chéo cạnh do tăng chiều rộng cạnh. Toán tử xói mòn sẽ loại bỏ tất cả các điểm biên khỏi một đối tượng và tất cả các điểm biên đều có chiều rộng là một pixel duy nhất. Nếu hình ảnh bị xói mòn bị trừ đi bởi hình ảnh gốc, kết quả sẽ là một cạnh rộng một pixel. Mỗi thành phần thu được, được gọi là một hộp cạnh (Edge Box - EB)

Các ký tự sẽ có kích thước xấp xỉ đồng nhất trong khu vực. Thuộc tính này được sử dụng để loại bỏ các EB không phải ký tự bằng cách phân tích nhị phân dựa trên phương pháp tiếp cận Cửa sổ trượt. Các bước liên quan đến thuật toán như sau:

a) Kích thước của các EB khác nhau được tính toán và lưu trữ trong mảng A [].

b) Ban đầu cửa sổ trên các phần tử mảng được giả định bằng cách bao phủ hai phần tử đầu tiên A [i] dưới dạng con trỏ trái (LP) và A [i + 1] dưới dạng con trỏ phải (RP).

c) Nếu sự khác biệt giữa LP và RP lớn hơn ngưỡng, chỉ xem xét RP và đặt nó là LP và trượt kích thước cửa sổ để che phần tử tiếp theo và kiểm tra sự khác biệt giữa hai phần tử đó.

d) So sánh lần lượt điểm đánh dấu cuối bên trái với các phần tử còn lại và nếu sự khác biệt < ngưỡng, mở rộng cửa sổ bằng cách đưa các phần tử đó vào cửa sổ và đóng băng cửa sổ khi điều kiện không thoả mãn. Cửa sổ này chứa các EB ký tự có kích thước thống nhất được xác định giữa LP và RP.

e) Sau đó, tính toán tối thiểu của các phần tử trong cửa sổ được đóng băng đầu tiên đó (MW1). Ở đây, giá trị tối thiểu được chọn để cho phép các ký tự có kích thước hỗn hợp.

f) Sau đó trượt cửa sổ để kiểm tra các phần tử còn lại và xác định các cửa sổ hợp lệ khác (Wi) và giá trị nhỏ nhất của các cửa sổ đó là MW1, MW2, … MWi

g) Tính toán tối thiểu MWi để cho kích thước hộp cạnh đồng nhất là ngưỡng Ts.

Bây giờ chia đôi hình ảnh với ngưỡng (Ts). Các EB có kích thước nhỏ hơn Ts bị loại bỏ. Pixel văn bản nền trước được hiển thị dưới dạng màu trắng và nền là pixel màu đen.

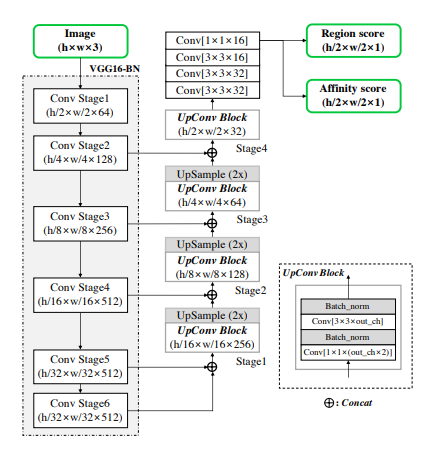
EB = 1 {Nếu kích thước (EB)> = Ts}

EB = 0 {Nếu kích thước (EB) <= Ts}

#### **1.2.2. Phương pháp học sâu với mô hình CRAFT (Character Region Awareness For Text)**

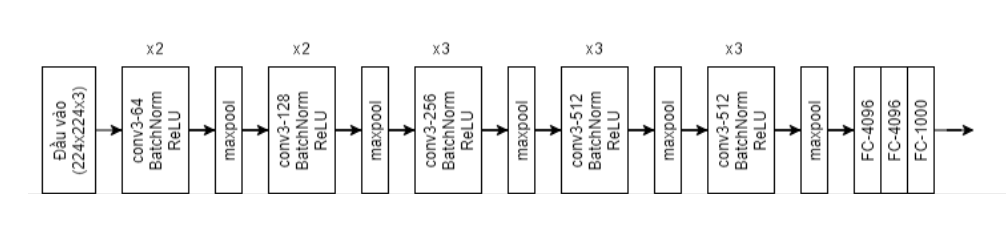
https://arxiv.org/pdf/1904.01941.pdf

Mô hình CRAFT là một mô hình phát hiện văn bản dựa trên phân đoạn ngữ nghĩa được đề xuất bởi nhóm tác giả đến từ Clova AI Research, NAVER Corp, được thiết kế với một mạng nơ-ron phức hợp trả về điểm số khu vực (region score) và điểm mối quan hệ (affinity score). Điểm số khu vực được sử dụng để nhận diện các ký tự riêng lẻ trong hình ảnh và điểm số quan hệ được sử dụng để nhóm từng ký tự thành một từ. Mô hình này là một kiến trúc mạng phức hợp dựa trên VGG-16 với chuẩn hóa theo mẻ (batch normalization) được sử dụng làm phần khung. Mô hình sử dụng kết nối bỏ qua trong phần giải mã, tương tự như U-Net ở chỗ nó tổng hợp các đặc điểm ở mức thấp.



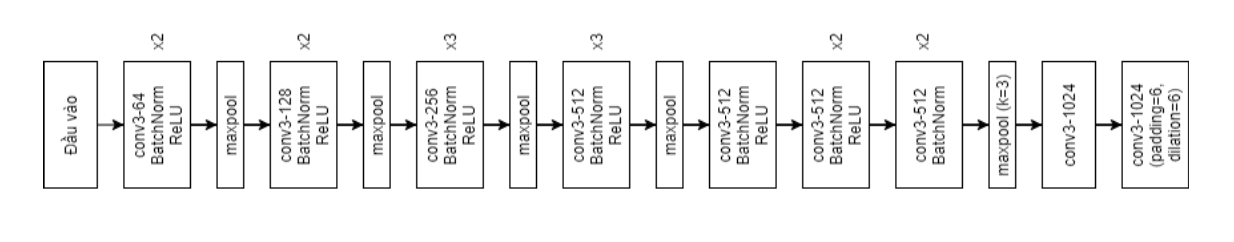
Hình 2.11: Kiến trúc mô hình CRAFT

Từ hình ta thấy phần khung trích xuất đặc trưng của CRAFT là mạng VGG16 Batch Normalization, mạng mà thêm vào giữa mỗi lớp tích chập và lớp maxpooling một lớp chuẩn hóa theo mẻ (batch norm) được thể hiện trên hình 2.6:



Hình 2.12: Kiến trúc mạng VGG16BN

VGG16 Batch Normalization áp dụng trong CRAFT có một số thay đổi ở điểm nó đã bỏ đi các lớp kết nối đầy đủ và xóa đi 1 lớp nhân chập ở nhóm cuối đồng thời thêm 1 lớp maxpool với kích thước bộ lọc là 3, tiếp đó là 1 lớp nhân chập conv3-1024 và cuối cùng là 1 lớp nhân chập conv3-1024 với vùng đệm bằng 6 và khoảng cách giữa các phần tử trong bộ lọc bằng 6 như trên hình 2.10.



Hình 2.13: Mạng VGG16BN áp dụng trong CRAFT

Kết quả đầu ra của mạng khung (gọi là phần thu hẹp) là một tensor có kích thước h/32 x 2/32 x 512 với kích thước được giảm đi nhưng độ sâu được tăng lên. Tiếp theo đến phần mở rộng gồm các layer đối xứng tương ứng với các layer của phần thu hẹp. Quá trình Upsampling với 1 khối mạng tích chập và chuẩn hóa được áp dụng giúp cho kích thước layer tăng dần lên. Cuối cùng đầu ra của mô hình là ma trận điểm vùng và ma trận ái lực đều có kích thước h/2 x w/2. Mỗi điểm số vùng thể hiện xác suất pixel đã cho là trung tâm của ký tự và điểm số mối quan hệ biểu thị xác suất của pixel là trung tâm của khoảng cách giữa các ký tự liền kề. Sử dụng một số phương pháp hậu xử lý để nhận được kết quả cuối cùng là các hộp giới hạn bao quanh mỗi kí tự, từ hay dòng tùy theo thiết lập.

### 

### **1.3. Các phương pháp nhận dạng chuỗi kí tự**

Phần này giới thiệu các phương pháp đang được sử dụng nhiều trong nhận dạng chuỗi kí tự như CRNN dựa trên hàm mất mát CTC và TBRA dựa trên phương pháp Attention.

#### **1.3.1. Phương pháp CRNN**

https://arxiv.org/pdf/1507.05717.pdf

Mô hình CRNN là một mô hình được đề xuất trong bài báo “*An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition*” bởi nhóm tác giả đến từ trường đại học và khoa học công nghệ Huazhong. CRNN là một kiến trúc mạng tích hợp các phần: trích xuất đặc trưng, mô hình trình tự và phiên mã thành vào một khung thống nhất. Theo tác giả, so với các hệ thống trước đây để nhận dạng văn bản cảnh, kiến ​​trúc CRNN sở hữu bốn đặc tính đặc biệt:

(1) Là một thuật toán được huấn luyện liền mạch từ đầu đến cuối, trái ngược với hầu hết các thuật toán hiện có mà các thành phần được huấn luyện và điều chỉnh riêng biệt.

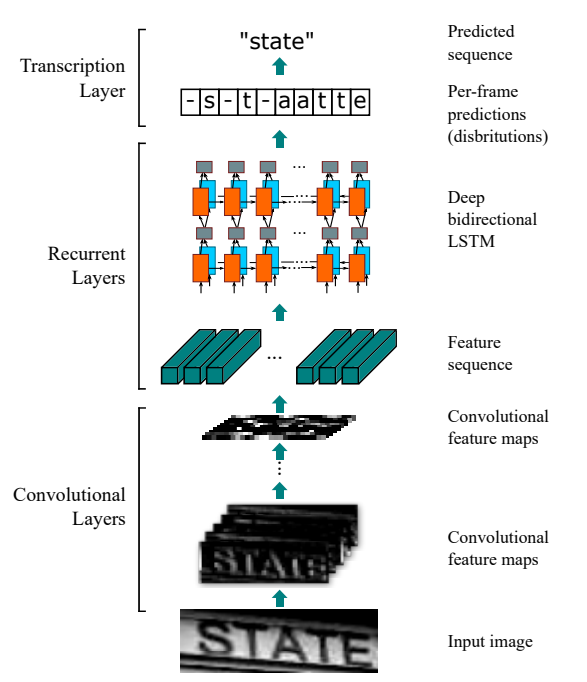
(2) Nó xử lý các chuỗi với độ dài tùy ý, không liên quan đến việc phân đoạn kí tự hoặc chuẩn hóa theo tỷ lệ ngang.

(3) Nó không bị giới hạn trong bất kỳ từ điền nào cho trước và đạt được hiệu suất tốt đối với những văn bản cấu thành từ từ vựng (lexincon-based) hay từ các từ viết tắt, từ đặc biệt (lexicon-free).

(4) Nó tạo ra một mô hình hiệu quả nhưng nhỏ hơn nhiều, dễ áp dụng hơn cho các tình huống ứng dụng trong thế giới thực.

Các thử nghiệm trên các điểm chuẩn tiêu chuẩn, bao gồm bộ dữ liệu IIIT-5K [40], các dữ liệu văn bản đường phố và ICDAR [41], chứng minh tính ưu việt của thuật toán được đề xuất so với các kỹ thuật trước đây.

Kiến trúc mạng của CRNN, như trong Hình 2.8, từ dưới lên bao gồm ba thành phần: các lớp tích chập (Convolutional Layers), các lớp hồi quy (Recurrent Layers) và một lớp phiên mã (Transcription Layer). Ở cuối CRNN, các lớp tích chập tự động trích xuất chuỗi đặc điểm từ mỗi hình ảnh đầu vào. Tiếp theo, một mạng hồi quy được xây dựng để đưa ra dự đoán cho từng khung của chuỗi đặc trưng. Lớp phiên mã ở đầu CRNN được sử dụng để dịch các dự đoán trên mỗi khung hình bởi các lớp hồi quy thành một chuỗi nhãn. Mặc dù CRNN bao gồm các loại kiến trúc mạng khác nhau (ví dụ: CNN và RNN), nó có thể được huấn luyện cùng với một hàm mất mát.

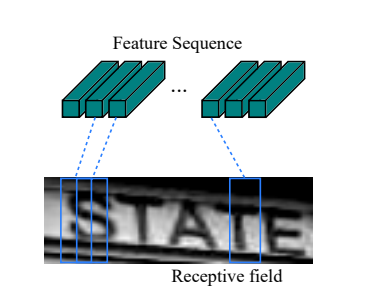


Hình 2.14: Kiến trúc CRNN [18]

##### **a. Trích xuất đặc trưng**

Trong mô hình CRNN, thành phần của các lớp tích chập được xây dựng bằng cách lấy các lớp tích chập và pooling từ mô hình CNN tiêu chuẩn (các lớp fully-connected sẽ bị loại bỏ). Những thành phần này sẽ được sử dụng để trích xuất một biểu diễn đặc trưng từ một hình ảnh đầu vào. Trước khi được đưa vào mạng, tất cả các hình ảnh cần được thu nhỏ đến cùng một kích thước. Sau đó, một chuỗi các vectơ đặc trưng được trích xuất từ ​​các bản đồ đặc trưng được tạo ra bởi thành phần của các lớp tích chập, là đầu vào cho các lớp hồi quy. Cụ thể, mỗi vectơ đặc trưng của chuỗi đối tượng được tạo từ trái sang phải kết nối với nhau theo các cột. Điều này có nghĩa là vectơ đặc trưng thứ i là sự ghép nối của các cột thứ i của cả bản đồ. Chiều rộng của mỗi cột được cố định thành một pixel. Khi các lớp của hàm tích chập, max-pooling và chức năng kích hoạt theo từng phần tử hoạt động trên các vùng cục bộ, chúng là bất biến. Do đó, mỗi cột của bản đồ đối tượng tương ứng với một vùng hình chữ nhật của hình ảnh gốc (được gọi là trường tiếp nhận) và các vùng hình chữ nhật đó có cùng thứ tự với các cột tương ứng của chúng trên bản đồ đối tượng từ trái sang phải Do đó, mỗi cột của bản đồ đặc trưng tương ứng với một vùng hình chữ nhật của hình ảnh gốc (được gọi là trường tiếp nhận) và các vùng hình chữ nhật đó có cùng thứ tự với các cột tương ứng của chúng trên bản đồ đối tượng từ trái sang phải. Như được minh họa trong hình 2.9, mỗi vectơ trong chuỗi đặc trưng được liên kết với một trường tiếp nhận và có thể được coi là bộ mô tả hình ảnh cho khu vực đó.

Do tính mạnh mẽ, phong phú và có thể huấn luyện nên các đặc trưng tích chập đã được áp dụng rộng rãi cho các bài toán nhận dạng hình ảnh khác nhau [19, 20]. Một số phương pháp tiếp cận trước đây đã sử dụng CNN để tìm hiểu cách biểu diễn mạnh mẽ cho các đối tượng dạng chuỗi như văn bản cảnh [21]. Tuy nhiên, các cách tiếp cận này thường trích xuất đại diện tổng thể của toàn bộ hình ảnh bởi CNN, sau đó các đặc điểm sâu cục bộ được thu thập để nhận ra từng thành phần của một đối tượng tương tự. Vì CNN yêu cầu các hình ảnh đầu vào phải được chia tỷ lệ thành một kích thước cố định để đáp ứng với kích thước đầu vào cố định của nó, nó không thích hợp cho các đối tượng dạng chuỗi do sự thay đổi độ dài lớn của chúng. Trong CRNN, các đặc trưng sâu được truyền vào các biểu diễn tuần tự để chúng trở nên bất biến với sự thay đổi độ dài của các đối tượng giống chuỗi.



Hình 2.15: Trường tiếp nhận. Mỗi vectơ trong chuỗi đặc trưng được trích xuất được liên kết với một trường tiếp nhận trên hình ảnh đầu vào và có thể được coi là vectơ đặc trưng của trường đó [18].

##### **b. Tạo chuỗi trình tự**

Như đã được đề cập tại phần giới thiệu, ở bước này áp dụng một mạng neuron hồi quy (Recurrent Neural Network - RNN) để tìm ra phân phối xác suất đầu ra từ các chuỗi đặc trưng tại bước trước. Trong đó mỗi đơn vị được sử dụng trong RNN là LSTM nên ta cũng có thể gọi là mạng LSTM.

\* Mạng LSTM

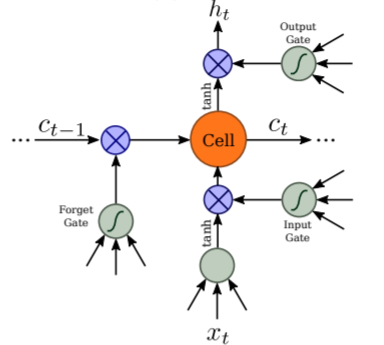
LSTM là một phiên bản mở rộng của mạng RNN, được đề xuất vào năm 1997 bởi Sepp Hochreiter và Jürgen Schmidhuber [27]. LSTM được thiết kế để giải quyết các bài toán về phụ thuộc xa (long-term dependencies) trong mạng RNN do bị ảnh hưởng bởi vấn đề Gradient biến mất.

https://towardsdatascience.com/the-vanishing-gradient-problem-69bf08b15484

\* Gradient biến mất: Khi nhiều layer sử dụng các hàm kích hoạt được thêm vào mạng neuron, các gradient của hàm mất mát sẽ tiến về 0, làm cho mạng khó đào tạo.

Một số hàm kích hoạt nhất định, chẳng hạn như hàm sigmoid, chuyển đổi một không gian đầu vào lớn thành một không gian đầu vào nhỏ có giá trị từ 0 đến 1. Do đó, một sự thay đổi lớn trong đầu vào của hàm sigmoid sẽ gây ra một thay đổi nhỏ trong đầu ra. Do đó, đạo hàm trở nên nhỏ. Đối với mạng nhỏ chỉ có một vài layer sử dụng các kích hoạt này, đây không phải là vấn đề lớn. Tuy nhiên, khi sử dụng nhiều layer hơn, nó có thể gây ra gradient quá nhỏ để quá trình đào tạo hoạt động hiệu quả.

Gradients của mạng neuron được tính bằng cách sử dụng lan truyền ngược (backpropagation). Nói một cách đơn giản, backpropagation tìm các dẫn xuất của mạng bằng cách di chuyển từng layer từ lớp cuối cùng đến lớp ban đầu. Theo quy tắc chuỗi, các dẫn xuất của mỗi lớp được nhân xuống mạng để tính các dẫn xuất của các layer ban đầu. Tuy nhiên, khi n layer ẩn sử dụng một kích hoạt như hàm sigmoid, n đạo hàm nhỏ sẽ được nhân với nhau. Do đó, gradient giảm theo cấp số nhân khi chúng ta truyền tới các lớp ban đầu. Một gradient nhỏ có nghĩa là trọng số và độ lệch của các lớp ban đầu sẽ không được cập nhật một cách hiệu quả. Vì các lớp ban đầu này thường rất quan trọng để nhận ra các yếu tố cốt lõi của dữ liệu đầu vào, nó có thể dẫn đến sự thiếu chính xác chung của toàn mạng. Có thể hiểu một cách đơn giản là mạng RNN cơ bản trong thực tế không có khả năng ghi nhớ thông tin từ các bước có khoảng cách xa và do đó những phần tử đầu tiên trong chuỗi đầu vào không có nhiều ảnh hưởng đến các kết quả tính toán dự đoán phần tử cho chuỗi đầu ra trong các bước sau.



Hình 2.16: Cấu trúc bên trong tế bào LSTM

Mạng LSTM có thể bao gồm nhiều tế bào LSTM (LSTM memory cell) liên kết với nhau và kiến trúc cụ thể của mỗi tế bào được biểu diễn như trong Hình 2.10. Bao gồm một ô nhớ và ba cồng: cổng vào, cổng ra và cổng quên.

https://nttuan8.com/bai-14-long-short-term-memory-lstm/

Ở state thứ t của mô hình LSTM:

- Output:  ta gọi c là cell state, h là hidden state.

- Input: trong đó  là input ở state thứ t của model; là output của layer trước.



Hình 2.17: Mô hình hoạt động của LSTM

​ tương ứng với **f**orget gate, **i**nput gate và **o**utput gate.

- Forget gate:

- Input gate:

- Output gate:

Với: 0 <, , < 1; , ,  là các hệ số bias.

Ta có:

, **forget gate** quyết định xem cần lấy bao nhiêu từ cell state trước và **input gate** sẽ quyết định lấy bao nhiêu từ input của state và hidden layer của layer trước.

, **output gate** quyết định xem cần lấy bao nhiêu từ cell state để trở thành output của hidden state. Ngoài ra  cũng được dùng để tính ra output  ​ cho state t.

Các kí hiệu được sử dụng trong các công thức trên:

+ là véc tơ đầu vào tại mỗi bước thời gian *t.*

*+* , , , , , , , là các ma trận trong mỗi tế bào LSTM.

+ , , , là các véc tơ độ lệch.

+ , , chứa các giá trị kích hoạt lần lượt cho các cổng quên, cổng vào, cổng ra.

+ , lần lượt là các véc tơ đại diện cho trạng thái nội bộ tế bào và giá trị ứng viên.

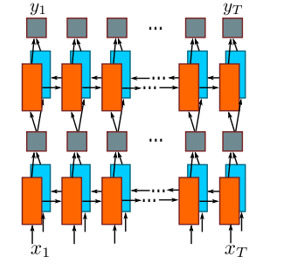
\* Áp dụng trong mô hình

Mạng neuron hồi quy hai chiều sâu (BiLSTM) được xây dựng để nhận đầu ra của các lớp tích chập. Các lớp hồi quy dự đoán phân phối nhãn cho mỗi khung trong chuỗi đặc trưng . Mạng này có những ưu điểm như sau :

- Thứ nhất, RNN có thế hiện khả năng mạnh mẽ trong việc nắm bắt thông tin theo ngữ cảnh. Sử dụng các dấu hiệu theo ngữ cảnh để nhận dạng chuỗi dựa trên hình ảnh ổn định và hữu ích hơn việc xử lý từng ký hiệu một cách độc lập. Bên cạnh đó, một số ký tự không rõ ràng sẽ dễ phân biệt hơn khi quan sát ngữ cảnh của chúng, ví dụ: dễ dàng nhận ra "il" bằng cách đối chiếu chiều cao của ký tự hơn là nhận ra từng ký tự một cách riêng biệt.

- Thứ hai, RNN có thể áp dụng thuật toán lan truyền ngược để có thể sửa những sai lệch từ đầu vào của nó, có thể thông qua các lớp tích chập, cho phép chúng ta cùng huấn luyện các lớp hồi quy và các lớp tích chập trong một mạng thống nhất.

- Thứ ba, RNN có thể hoạt động trên các chuỗi có độ dài tùy ý.



Hình 2.18. Cấu trúc của LSTM hai chiều sâu.

Kết hợp LSTM xuôi (trái sang phải) và ngược (phải sang trái).

Một đơn vị RNN truyền thống có một lớp ẩn tự kết nối giữa các lớp đầu vào và đầu ra của nó. Mỗi khi nó nhận được một khung trong chuỗi, nó sẽ cập nhật trạng thái ẩn của nó bằng một hàm phi tuyến tính lấy cả đầu vào hiện tại và trạng thái trong quá khứ làm đầu vào của nó . Sau đó, dự đoán được thực hiện dựa trên . Bằng cách này, các ngữ cảnh trong quá khứ được ghi lại và sử dụng để dự đoán. Tuy nhiên, như đã được đề cập, đơn vị RNN truyền thống gặp phải vấn đề Gradient biến mất, giới hạn phạm vi bối cảnh mà nó có thể lưu trữ và thêm gánh nặng cho quá trình huấn luyện nên tác giả đã sử dụng LSTM, cho phép nó nắm bắt các phụ thuộc xa, thường xảy ra trong các chuỗi dựa trên hình ảnh. Từ định hướng là việc sử dụng LSTM, tác giải nhận thấy trong trình tự dựa trên hình ảnh, bối cảnh từ cả hai hướng đều hữu ích và bổ sung cho nhau. Do đó, tác giả kết hợp hai LSTM, một chiều xuôi một chiều ngược, thành một LSTM hai chiều. Hơn nữa, nhiều LSTM hai chiều có thể được xếp chồng lên nhau, dẫn đến một LSTM hai chiều sâu như minh họa trong Hình 2.14. Cấu trúc sâu cho phép mức độ trừu tượng cao hơn cấu trúc nông và đã đạt được những cải tiến hiệu suất đáng kể trong nhiệm vụ nhận dạng giọng nói [13].

##### **c. Sự phiên mã**

Phiên mã là quá trình chuyển đổi các dự đoán trên mỗi khung hình do RNN trả về thành một chuỗi trình tự. Về mặt toán học, phiên mã là để tìm chuỗi trình tự có xác suất cao nhất được điều chỉnh dựa trên các dự đoán trên mỗi khung hình. Trong thực tế, tồn tại hai hình thức phiên mã, đó là phiên mã không có từ vựng và dựa trên từ vựng. Ở chế độ không có từ vựng, các dự đoán được thực hiện mà không có bất kỳ từ vựng nào. Trong chế độ dựa trên từ vựng, các dự đoán được thực hiện bằng cách chọn chuỗi nhãn có xác suất cao nhất.

\* Xác suất của chuỗi trình tự

Tác giả áp dụng xác suất có điều kiện được xác định trong Phân loại theo thời gian kết nối (Connectionist temporal classification - CTC) do Graves và các cộng sự đề xuất [24]. Xác suất có điều kiện của nhãn trình tự L được xác định dựa trên các dự đoán trên mỗi khung hình , và nó bỏ qua vị trí của mỗi nhãn trong L. Do đó, khi chúng ta sử dụng hàm negative log-likelihood của xác suất này làm hàm mất mát để huấn luyện mạng, chúng ta chỉ cần hình ảnh và chuỗi nhãn tương ứng của chúng, tránh tốn công gắn nhãn vị trí của các ký tự riêng lẻ.

Giả sử đầu ra của mạng là một chuỗi trong đó T là độ dài của chuỗi, trong đó với là xác suất lớn nhất của một phân phối xác suất trên tập nhãn kí tự và là xác suất của kí tự thứ k tại thời điểm t, trong đó L chứa toàn bộ kí tự nhãn và kí tự ‘blank’ là kí tự trống được thêm vào. Gọi là một chuỗi thuộc tập , các chuỗi có độ dài T với mỗi kí tự thuộc . Xác suất có điều kiện của mỗi chuỗi , được gọi là chuỗi căn chỉnh (alignment), với chuỗi đặc trưng đầu vào *x* là tích của các biến cố trong (giả định các biến cố độc lập nhau):

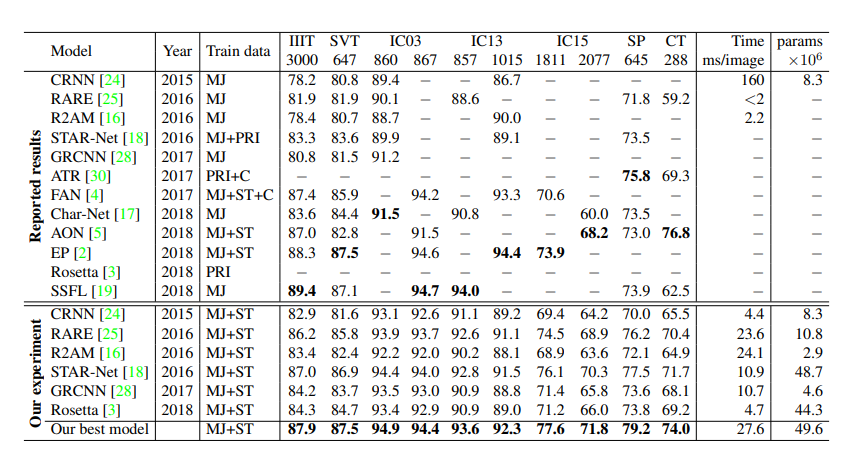
Ví dụ các chuỗi căn chỉnh từ nhãn “hello” là “heee---llo”, “-hell--loo”… Tiếp theo, tác giả định nghĩa hàm ánh xạ B : , trong đó là tập các chuỗi trình tự đầu ra có độ dài nhỏ hơn hoặc bằng T sau khi đã loại bỏ các kí tự lặp lại và kí tự “blank”, ví dụ . Xác suất có điều kiện của kết quả *l* (labelling) với đầu vào *x* được định nghĩa là tổng xác suất có điều kiện của các chuỗi trình tự với đầu vào *x* qua hàm B được kết quả *l*

(1)

Kết quả đầu ra thỏa mãn công thức:

#### **1.3.2. Phương pháp TRBA**

Năm 2019 nhóm tác giả đến từ Clova AI Research đã công bố bài báo “*What Is Wrong With Scene Text Recognition Model Comparisons? Dataset and Model Analysis*” [28] đề cập về sự không nhất quán của các tập dữ liệu huấn luyện và đánh giá cho những mô hình nhận dạng văn bản cảnh (scene text recognition) được đưa ra những năm gần đây mà có kết quả nhận dạng kém. Và họ đã giới thiệu một khuôn mẫu thống nhất bao gồm bốn bước áp dụng cho các mô hình nhận dạng văn bản cảnh. Họ cũng đưa ra đánh giá về nhiều mô hình như CRNN [18], RARE [25], R2AM [23], STAR-Net [30], GRCNN [31], Rosetta [32],…và mô hình TRBA (TPS-ResNet-BiLSTM-Attn) do họ đề xuất, mô hình nhận dạng văn bản cảnh (STR) có độ chính xác cao nhất trên các tập dữ liệu được kiểm nghiệm trên hình 2.13.



Hình 2.19: So sánh hiệu quả của các mô hình STR hiện có và thực nghiệm trên các mô hình đó của nhóm tác giả [28]

Các bước mà nhóm tác giả đề cập bao gồm:

- Chuyển đổi: chuẩn hóa hình ảnh chuỗi kí tự đầu vào bằng cách sử dụng mạng biến áp không gian (STN) để nắn những chuỗi kí tự bị cong, nghiêng trên ảnh.

- Trích xuất đặc trưng: sử dụng mạng nơ ron tích chập để ánh xạ hình ảnh đầu vào thành những đặc trưng mà liên quan đến nhận dạng kí tự, đồng thời loại bỏ đồng thời loại bỏ các đặc trưng không liên quan như phông chữ, màu sắc, kích thước và nền.

- Tạo trình tự: sử dụng mạng nơ ron hồi quy (RNN) để nắm bắt thông tin ngữ cảnh trong một chuỗi ký tự cho bước tiếp theo để dự đoán từng ký tự một cách tốt hơn, thay vì chỉ sử dụng chuỗi đặc trưng từ bước trước đó.

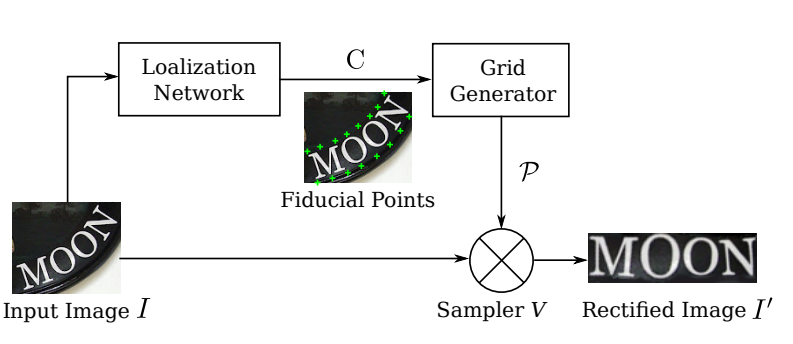
- Dự đoán: dự đoán chuỗi ký tự đầu ra từ các đặc điểm đã xác định của hình ảnh.

##### **a. Bước chuyển đổi**

Giai đoạn này biến hình ảnh đầu vào thành hình ảnh được chuẩn hóa . Các hình ảnh chuỗi kí tự thường có nhiều hình dạng cong, nghiêng khác nhau. Nếu những hình ảnh đầu vào như vậy không được xử lý thì giai đoạn trích xuất trích xuất đặc trưng tiếp theo cần phải học cách biểu diễn của các chuỗi kí tự với nhiều hình dạng khác nhau. Để giảm bớt gánh nặng này, phép biến đổi spline bản mỏng (TPS), đã được áp dụng để chuẩn hóa hình ảnh đầu vào.

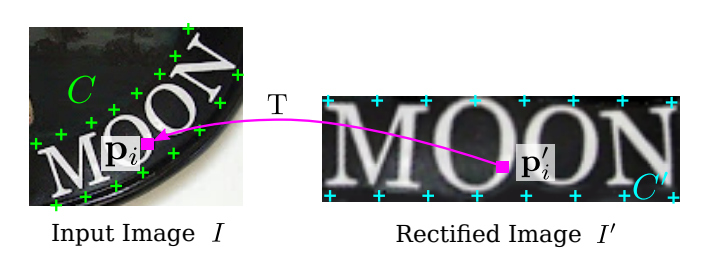
\* Mạng STN với phép biến đổi TPS

STN [22], được đề xuất bởi Max Jaderberg và các cộng sự năm 2015, là một kiến trúc mạng bao gồm một chuỗi các quy trình: tìm đường biên văn bản, liên kết vị trí của các pixel trong đường biên với vị trí của hình ảnh chuẩn hóa và tạo hình ảnh chuẩn hóa bằng cách sử dụng các giá trị của pixel và thông tin liên kết. Các quy trình như vậy được gọi tương ứng là mạng cục bộ hóa, bộ tạo lưới và bộ lấy mẫu hình ảnh và được thể hiện trên hình…. Về mặt khái niệm, TPS sử dụng phép nội suy đường thẳng giữa một tập hợp các điểm cơ sở (fiducial points) đại diện cho đường viền xung quanh của chuỗi kí tự trong một hình ảnh, số lượng các điểm cơ sở là F và nó là một hằng số.



Hình 2.20: Kiến trúc của STN. Mạng cục bộ hóa định vị các điểm cơ sở C, bộ tạo lưới tạo ra lưới lấy mẫu P, bộ lấy mẫu V tạo hình ảnh chuẩn hóa I' từ hình ảnh đầu vào I và P [25]

Mạng cục bộ hóa là một mạng CNN bao gồm các lớp tích chập, các lớp tổng hợp và lớp kết nối đầy đủ. Lớp đầu ra của mạng, lớp kết nối đầy đủ cuối cùng, được thiết lập với số lượng đầu ra là 2F sau đó được đưa ra hàm kích hoạt *tanh* để đưa các giá trị đầu ra nằm trong [-1, 1], F là số điểm cơ cở trên hình ảnh đầu vào và các điểm cơ sở được biểu thị bởi ma trận trong đó cột thứ f chứa tọa độ của điểm cơ sở thứ f. Mạng cục bộ hóa định vị các điểm cơ sở trên toàn ảnh, nó được mong đợi có thể nắm bắt hình dạng chuỗi kí tự tổng thể của hình ảnh đầu vào và khoanh vùng các điểm cở sở cho phù hợp.



Hình 2.21: Tập điểm cơ sở và bộ chuyển đổi TPS [25]

Bộ tạo lưới tính toán những tham số chuyển đổi TPS và tạo ra lưới lấy mẫu. Đầu tiên định nghĩa trước một tập các điểm cơ sở của hình ảnh chuẩn hóa biểu thị bởi ma trận . Như được minh họa trên hình … bộ tạo lưới cung cấp một hàm ánh xạ các vùng được xác định ở mạng cục bộ hóa và hình ảnh chuẩn hóa. Hàm ánh xạ được tham số hóa bởi ma trận T

(1)

trong đó là một ma trận được xác định bởi C, do đó nó cũng là hằng

(2)

trong đó phần tử hàng i cột j của R có giá trị , là khoảng cách euclidean giữa và . Lưới pixels trên ảnh chuẩn hóa được biểu thị bằng , trong đó là tọa độ của pixel thứ i, N là số lượng pixel. Trên hình 3, với mỗi pixel trong , thực hiện tìm điểm trong *I* bằng việc áp dụng phép chuyển đổi:

(3)

(4)

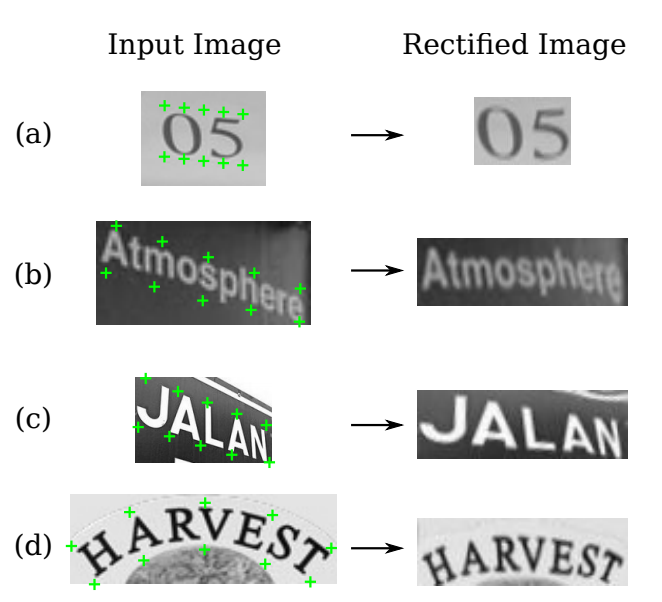
(5)

trong đó là khoảng cách euclidean giữa và . Thực hiện lặp lại công thức (3) trên toàn bộ các điểm của , tạo ra lưới trên *I.*

Cuối cùng, trong bộ lấy mẫu, những giá trị pixel của được nội suy song tuyến từ các điểm ảnh gần trên hình ảnh đầu vào. Thực hiện tính toán tất cả các giá trị điểm ảnh, nhận được hình ảnh đầu ra :

trong đó V đại diện cho bộ lấy mẫu song tuyến.

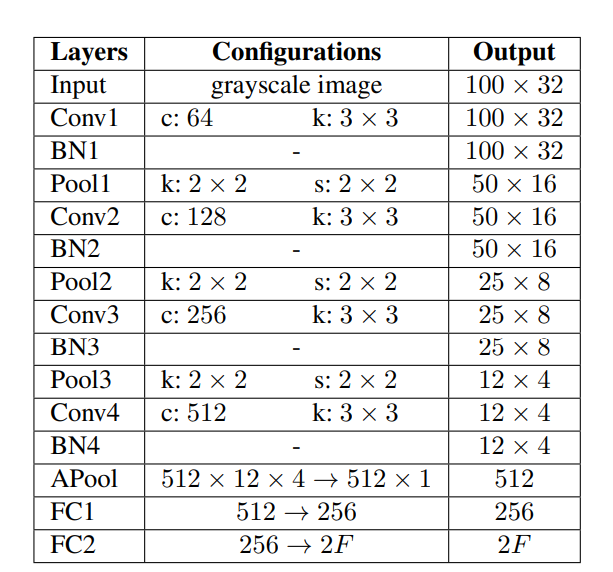
Tính linh hoạt của phép chuyển đổi TPS cho phép biến đổi các hình ảnh chuỗi kí tự không đúng chuẩn thành các hình ảnh chỉnh sửa có chứa chuỗi kí tự đúng chuẩn. Trong hình 2.18 là một số loại chuỗi kí tự không đúng chuẩn phổ biến, bao gồm a) chuỗi kí tự có hộp giới hạn không cắt sát, do trình phát hiện văn bản chưa tốt; b) chuỗi kí tự nhiều hướng, gây ra bởi các camera chụp không nằm ngang; c) văn bản phối cảnh, do góc máy ảnh xem bên; d) Chữ cong, một trường hợp thường thấy. STN có thể chỉnh sửa các hình ảnh chứa các loại chuỗi kí tự bất thường này, làm cho chúng dễ đọc hơn đối với trình nhận dạng sau.



Hình 2.22: STN căn chỉnh một số hình ảnh chuỗi kí tự thường gặp [25]

\* Áp dụng trong mô hình

STN yêu cầu mạng cục bộ tính toán các điểm cơ sở của hình ảnh đầu vào. Nhóm tác giả thiết kế mạng cục bộ bằng cách tuân theo hầu hết các thành phần của các nghiên cứu trước đó [22] và thêm các lớp chuẩn hóa theo mẻ và tổng hợp trung bình thích ứng (adaptive average pooling) để ổn định việc huấn luyện mạng. Hình 2.19 thể hiện kiến trúc mạng cục bộ bao gồm 4 lớp tích chập, mỗi lớp theo sau là lớp chuẩn hóa theo mẻ và lớp tổng hợp tối đa 2 x 2. Kích thước bộ lọc, kích thước phần đệm và bước nhảy tương ứng là 3, 1, 1 cho tất cả các lớp tích chập. Theo sau lớp chập cuối cùng là lớp gộp trung bình thích ứng. Sau đó, hai lớp được kết nối đầy đủ như sau: 512 đến 256 và 256 đến 2F. Đầu ra cuối cùng là véc tơ 2F chiều tương ứng với giá trị của tọa độ x, y của F điểm cơ sở trên hình ảnh đầu vào. Các hàm kích hoạt cho tất cả các lớp là ReLU



Hình 2.23: Kiến trúc của mạng cục bộ trong STN [28]

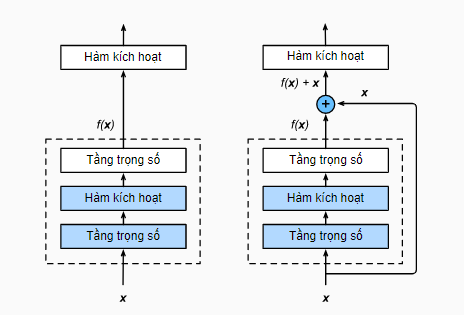
##### **b. Bước trích xuất đặc trưng**

Trong giai đoạn này, một mạng nơ ron tích chập sẽ trích xuất từ hình ảnh đầu vào và kết quả là một bản đồ đặc trưng (I là số lượng cột của bản đồ đặc trưng). Mỗi cột trong bản đồ đặc trưng kết quả được thực hiện bằng một trình trích xuất đặc trưng có một trường tiếp nhận có thể phân biệt được tương ứng dọc theo đường ngang của hình ảnh đầu vào. Các đặc trưng này được sử dụng để ước tính ký tự trên mỗi trường tiếp nhận. Nhóm tác giả nghiên cứu ba kiến ​​trúc VGG [17], RCNN [23], và ResNet [29]. VGG ở dạng ban đầu bao gồm nhiều lớp chập, tiếp theo là một vài lớp được kết nối đầy đủ. RCNN là một biến thể của CNN có thể được áp dụng đệ quy để điều chỉnh các trường tiếp nhận của nó tùy thuộc vào hình dạng ký tự. ResNet là CNN với các kết nối dư giúp giảm bớt việc huấn luyện các mạng CNN sâu hơn.

Trong phạm vi đồ án này sử dụng mạng ResNet tương ứng phần R trong mô hình TRBA.

\* Mạng ResNet

ResNet (Residual Network) [29] được giới thiệu vào năm 2015 và thậm chí đã giành được vị trí thứ 1 trong cuộc thi ILSVRC 2015 với tỉ lệ lỗi top 5 chỉ 3.57%. Không những thế nó còn đứng vị trí đầu tiên trong cuộc thi ILSVRC and COCO 2015 với ImageNet Detection, ImageNet localization, Coco detection và Coco segmentation. Hiện tại thì có rất nhiều biến thể của kiến trúc ResNet với số lớp khác nhau như ResNet-18, ResNet-34, ResNet-50, ResNet-101, ResNet-152...Với tên là ResNet theo sau là một số chỉ kiến trúc ResNet với số lớp nhất định. Mạng ResNet được thiết kế để làm việc với hàng trăm hoặc hàng nghìn lớp chập. Một vấn đề xảy ra khi xây dựng mạng CNN với nhiều lớp chập sẽ xảy ra hiện tượng Gradient biến mất dẫn tới quá trình học tập không tốt. Cho nên giải pháp mà ResNet đưa ra là sử dụng kết nối "tắt" đồng nhất để xuyên qua một hay nhiều lớp. Một khối như vậy được gọi là một Residual Block, như trong hình sau:

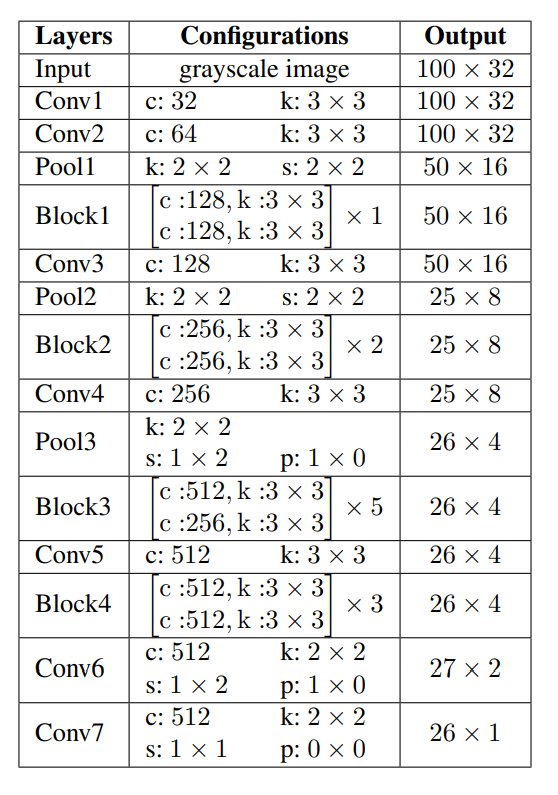


Hình 2.24: Sự khác biệt giữa một khối thông thường (trái) và một khối phần dư (phải) [44]

Từ mạng nơ ron trên hình 2.20, ký hiệu đầu vào là . Giả sử ánh xạ lý tưởng muốn học được là  và được dùng làm đầu vào của hàm kích hoạt. Phần nằm trong viền nét đứt bên trái phải khớp trực tiếp với ánh xạ . Điều này có thể không đơn giản nếu ta không cần khối đó và muốn giữ lại đầu vào x. Khi đó, phần nằm trong viền nét đứt bên phải chỉ cần tham số hoá độ lệch khỏi giá trị x, bởi vì ta đã trả về . Trên thực tế, ánh xạ phần dư thường dễ tối ưu hơn, vì chỉ cần đặt . Nửa bên phải mô tả khối phần dư cơ bản của ResNet.

\* Áp dụng trong mô hình

Nhóm tác giả triển khai mô đun ResNet với 29 lớp có thể huấn luyện. Trên hình 2.21 mô tả chi tiết về mạng được sử dụng, trong đó c là số kênh và k là kích thước bộ lọc và s là bước nhảy. Mỗi lớp Conv gồm các lớp theo thứ tự: 1 lớp tích chập, 1 lớp BatchNorm và 1 lớp ReLU. Mỗi lớp Block là các khối phần dư gồm các lớp theo thứ tự: 1 lớp tích chập , 1 lớp BatchNorm, 1 lớp relu, 1 lớp tích chập và 1 lớp BatchNorm rồi sau đó đầu ra của chuỗi lớp này được kết hợp với đầu vào ở lớp trước lớp Block rồi qua 1 lớp ReLU. Với đầu vào là ảnh xám có kích thước 100 x 32 tương ứng chiều rộng là 100 và chiều cao là 32 pixel thì đầu ra có kích thước là 26 x 1.



Hình 2.25: Mạng ResNet được sử dụng trong TBRA [28]

##### **c. Bước tạo trình tự**

Giai đoạn này giúp nắm bắt thông tin ngữ cảnh trong một chuỗi ký tự cho giai đoạn tiếp theo từ chuỗi đặc trưng V, được tạo thành từ bản đồ đặc trưng ở giai đoạn trích xuất đặc trưng, để dự đoán từng ký tự một cách tốt hơn thay vì làm điều đó một cách độc lập.

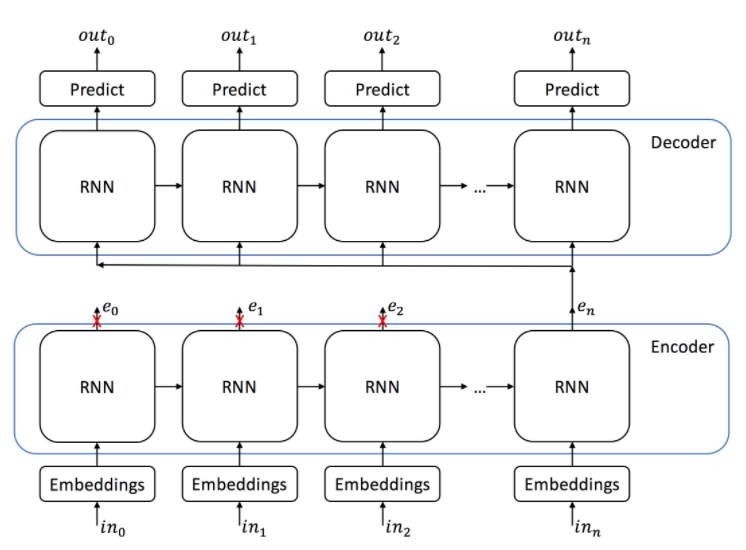
\* Áp dụng trong mô hình

Sử dụng 2 mạng LSTM hai chiều (BiLSTM), cũng được sử dụng trong CRNN [18] đã được đề cập. Mạng BiLSTM thứ *l* xác định hai trạng thái ẩn , được tính toán qua chuỗi trình tự và đảo ngược của nó. Để thu được đầu ra là phân phối xác suất của các các véc tơ đặc trưng tại bước trước.

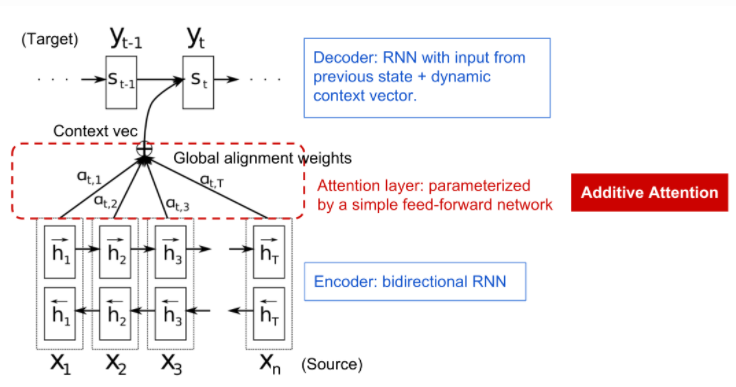
##### **d. Bước dự đoán kết quả**

Giai đoạn này kết hợp với giai đoạn tạo trình tự để tính toán kết quả đầu ra của chuỗi kí tự từ những đặc trưng của hình ảnh đã được xác định trước đó. Tại bước này tác giả triển khai bộ giải mã dựa trên cơ chế chú ý (attention-based decoder). Bộ giải mã nhằm tìm phân phối xác suất của các đặc trưng học tập (learning features) ở bước mã hóa, được tính toán qua mạng RNN tại bước tạo trình tự, từ đó xác định đâu là nhãn đầu ra. Mô hình mã hóa – giải mã trên thuộc lớp seq2seq tức ánh xạ chuỗi thành chuỗi.

Cơ chế chú ý (attention) được giới thiệu trong bài báo “Neural machine translation by jointly learning to align and translate” bởi Dzmitry Bahdanau và các cộng sự năm 2015 [33], nó được áp dụng đầu tiên trong mô hình dịch máy (Neural Machine Translation), theo tác giả các kí tự ở đầu vào sẽ có mối liên hệ lớn hơn đối với kí tự ở đầu ra cùng vị trí. Do đó attention hiểu một cách đơn giản sẽ giúp mô hình điều chỉnh sự tập trung lớn hơn ở các cặp từ (input, output) nếu chúng có vị trí tương đương hoặc gần tương đương. Với mô hình seq2seq thông thường thì việc mã hóa toàn bộ thông tin từ đầu vào (source) vào 1 vector cố định khiến mô hình khi thực hiện trên các câu dài (long sentence) không thực sự tốt, mặc dù sử dụng LSTM (BiLSTM, GRU [34]) để khắc phục điểm yếu của mạng RNN truyền thống với hiện tượng Gradient biến mất, nhưng như thế có vẻ vẫn chưa đủ, đặc biệt đối với những câu dài hơn những câu trong tập dữ liệu huấn luyện. Từ đó, trong bài báo, tác giả đề xuất 1 cơ chế cho phép mô hình có thể chú trọng vào những phần quan trọng và thay vì chỉ sử dụng véc tơ ngữ cảnh (context vector) được tạo ra từ lớp cuối cùng của bộ mã hóa, tác giả sử dụng tất cả các đầu ra của từng tế bào (cell) qua từng bước thời gian (timestep), kết hợp với trạng thái ẩn của từng tế bào để "tổng hợp" ra một véc tơ ngữ cảnh (attention vector) và dùng nó làm đầu vào cho từng tế bào trong bộ giải mã.



Hình 2.26: Cơ chế seq2seq phổ biến (không có attention) [63]



Hình 2.27: Cơ chế seq2seq có attention [63]

Nội dung cơ chế, với đầu vào có độ dài *n* và đầu ra có độ dài *m*, mỗi xác suất của kí tự đầu ra sẽ phụ thuộc vào kí tự trước đó , trạng thái ẩn của bộ giải mã và véc tơ ngữ cảnh theo công thức:

trong đó , là các tham số huấn luyện, trạng thái ẩn được tính thông qua mạng LSTM trong bộ giải mã với thông tin từ trạng thái ẩn trước đó , kí tự kết quả trước đó và véc tơ ngữ cảnh

Véc tơ ngữ cảnh được tính theo công thức:

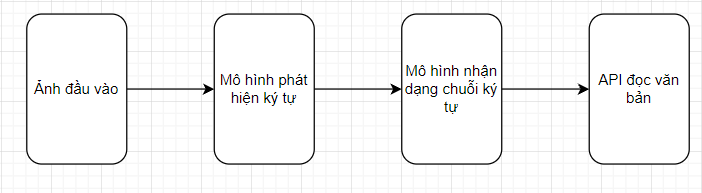
Với là trạng thái ẩn của bộ mã hóa, mỗi được gọi là điểm chú ý (attention score), được tính thông qua 1 hàm softmax là trọng số riêng được tính tại timestep *t* ứng với mỗi trạng thái ẩn của bộ mã hóa ​. Sử dụng hàm softmax như là một cách để chuẩn hóa đối với từng alignment score hay attention energy ​ thể hiện mức độ tương quan giữa và :

được gọi là mô hình căn chỉnh (alignment model) với mục đích đánh giá mức độ tương quan của của kí tự thứ *j* ở bộ mã hóa và kí tự thứ *i* ở bộ giải mã, có nhiều cách tính như Content-base Attention [35], Additive Attention [33], Multiplicative Attention [37], Dot Product [37],… Ở mô hình TBRA tác giả sử dụng Additive Attention:

trong đó , và là các ma trận trọng số.

## **2. Xây dựng mô hình**

Từ cơ sở lý thuyết ở trên, đồ án sẽ đề xuất mô hình hoàn chỉnh với nhiệm vụ chuyển đổi văn bản thành giọng nói, là sự kết hợp của những mô hình được huấn luyện riêng biệt.



Hình 2.28.Hệ thống chuyển đổi văn bản thành giọng nói

\* Phát hiện ký tự

Đầu vào của anh trong phạm vi đồ án được giả định đã căn chỉnh tương đối chuẩn xác với nền không phúc tạp. Mô hình CRAFT được huấn luyện trước của nhóm tác giả sẽ được áp dụng chop ha tiền xử lý. Hình ảnh đầu vào sau khi qua xử lý sẽ thu về kết quả là một danh sách các hộp giới hạn hình chữ nhật bao quanh mỗi từ, kết hợp với hàm để sắp xếp thứ tự các hộp đúng với thứ tự các chữ từ trái sang phải và theo từng dòng.

\* Nhận dạng chuỗi kí tự

Mô hình TBRA được sử dụng để dự đoán các từ từ ảnh đầu vào. Cụ thể như sau: đầu tiên những hình ảnh ở mức các từ được cắt từ các hộp giới hạn ở bước trước được đưa qua mô hình TRBA để thu được đầu ra có dạng một chuỗi ký tự s. Từ kết quả của ảnh đầu vào, một API sẽ được sử dụng để đọc nội dung dự đoán và phát cho người sử dụng.

## **3. Kết luận**

Chương 2 đã trình bày tổng quan về các phương pháp hiện có của từng bước trong bài toán nhận dạng và chuyển đổi văn bản thành giọng nói bao gồm các phương pháp tiền xử lý, phát hiện vùng kí tự và nhận dạng kí tự.

Ở chương tiếp theo, đồ án sẽ tiến hành xây dựng hệ thống chuyển đổi văn bản thành âm thanh áp dụng trong các thư viện đồng thời tiến huấn luyện các mô hình và thử nghiệm với hệ thống đã xây dựng.

# CHƯƠNG 3: ĐỀ XUẤT GIẢI PHÁP ỨNG DỤNG

## **1. Kịch bản hệ thống**

Đồ án đề xuất xây dựng hệ thống chuyển văn bản sách giấy và chuyển thành giọng nói để hỗ trợ người khiếm thị đọc sách tại các thư viện.

Kịch bản hệ thống gồm các bước như sau:

Bước 1: Huấn luyện các mô hình

Bước này được thực hiện trong lúc xây dựng hệ thống, các mô hình được huấn luyện từ những bộ dữ liệu văn bản tiếng Việt được thu thập trước đó hoặc tự sinh theo các tham số về font chữ, màu nền.

Bước 2: Trích xuất thông tin

Khi độc giả muốn đọc sách, nhân viên thư viện sẽ hỗ trợ thiết đặt sách vào vị trí phù hợp của camera, hình ảnh trang sách sẽ được chụp lại. Hình ảnh sẽ được xử lý để lấy các hộp giới hạn của từ và sắp xếp lại theo thứ tự lần lượt để tạo thành một đoạn văn có nghĩa. Sau đó hình ảnh và danh sách các hộp giới hạn được gửi sang server xử lý nhận diện

Bước 3: Nhận diện từ tiếng Việt

Với hình ảnh và toạ độ của các hộp giới hạn, server nhận diện sẽ dự đoán các từ Nội dung sau khi được nhận diện sẽ gửi lại và đọc thành tiếng cho độc giả.

## **2. Xây dựng hệ thống**

### **2.1. Phân tích hệ thống**

Từ kịch bản đề xuất ở trên, đồ án xây dựng hệ chuyển đổi văn bản thành giọng nói như hình dưới

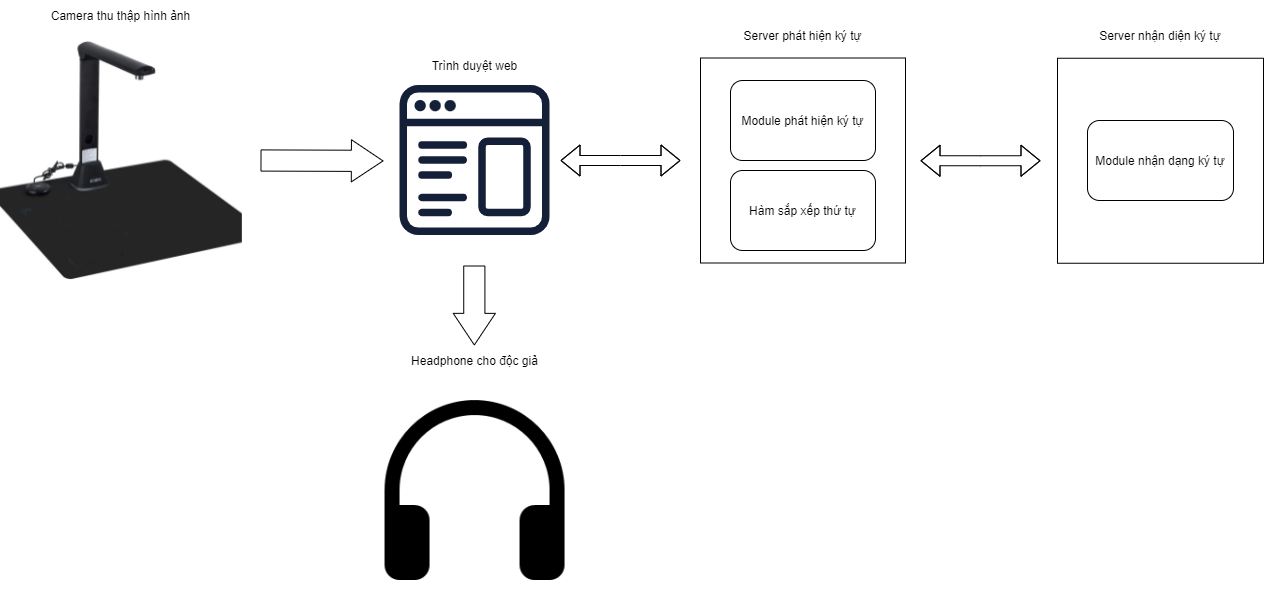
Hình 3.1: Biểu đồ ca sử dụng trong hệ thống

Hệ thống có 1 tác nhân chính:

* Người sử dụng: Bao gồm thủ thư và độc giả:
  + Thủ thư sẽ giúp đỡ độc giả chọn sách, đặt sách vào đúng vị trí phù hợp tại máy scan và hướng dẫn người đọc sử dụng (vị trí nút chụp hình, âm lượng…)
  + Độc giả sẽ bấm nút chụp và nghe nội dung qua tai nghe, khi kết thúc trang sách có thể tự mình lật trang và bấm nút chụp để tiếp tục nghe nội dung sách.

### **2.2. Thiết kế hệ thống**

Từ kịch bản trên, đồ án đề xuất xây dựng hệ thống nhận dạng trích xuất thông tin chứng minh thư nhân dân áp dụng trong các quầy giao dịch. Hệ thống dựa trên kiến trúc nhiều tầng, bao gồm 5 thành phần chính như hình 24.



Hình 3.2: Kiến trúc chung của hệ thống

* **Camera thu thập hình ảnh:** Thực hiện thu thập hình ảnh của trang sách.
* **Trình duyệt web:** Nhận ảnh được gửi từ người dùng, và gửi xử lý/trả kết quả dưới dạng tệp âm thanh phát trực tiếp.
* **Máy chủ phát hiện ký tự:**Là máy chủ với nhiệm vụ xử lý ảnh đầu vào, phát hiện các từ và sắp xếp lại thứ tự thành đoạn văn có nghĩa.
* **Máy chủ nhận dạng ký tự:** Là máy chủ cốt lõi của hệ thống, làm nhiệm vụ nhận ảnh từ máy chủ phát hiện ký tự, dự đoán các từ và trả về nội dung dự đoán của trang sách được đưa vào.

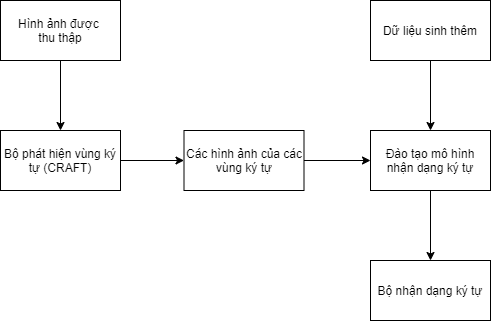
Các thành phần trên sẽ thực hiện 2 nhóm chức năng chính: Tiếp nhận/ trả thông tin và huấn luyện, nhận dạng. Mô đun Tiếp nhận/ trả thông tin chịu trách nhiệm các chức năng vào ra: Nhận ảnh từ camera, gửi xử lý và nhận xâu ký tự sau nhận diện để chuyển đổi thành giọng nói qua Google API. Còn module Huấn luyện và nhận dạng thực hiện các công việc chính liên quan đến thu thập, huấn luyện dữ liệu, nhận dạng văn bản… Trong phần tiếp theo, đồ án sẽ trình bày chi tiết mô đun Huấn luyện dữ liệu và Nhận dạng văn bản tiếng Việt.

## **3. Huấn luyện mô hình và nhận dạng trích xuất thông tin**

Phần này bao gồm hai quy trình được thực hiện theo vòng tuần hoàn. Mô hình nhận dạng trích xuất thông tin được huấn luyện trước với tập dữ liệu văn bản tiếng Việt được thu thập trước đó, kết hợp với bộ dự liệu được sinh theo thuật toán với các thông số cho trước về font chữ, nền ảnh.

### **3.1. Huấn luyện mô hình**

Việc huấn luyện các mô hình trong hệ thống bao gồm 2 pha như trên hình, được thực hiện bởi người quản trị. Hai pha này có thể thực hiện riêng biệt, không liên tục.

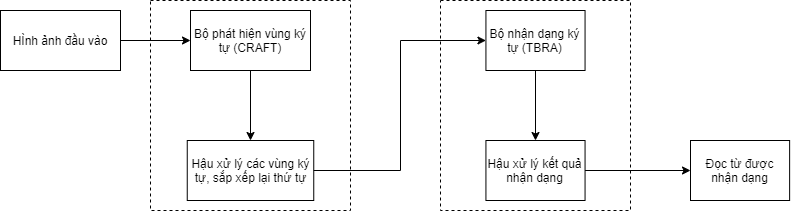


Hình 3.3: Quy trình huấn luyện mô hình

Quy trình thực hiện huấn luyện bộ nhận dạng kí tự, ảnh được đưa qua mô hình phát hiện vùng kí tự CRAFT, đã được huấn luyện trước (pretrained model) của nhóm tác giả với các tập dữ liệu gồm: tập dữ liệu sinh, IC13 và IC17, thu được ma trận điểm và ma trận ái lực. Với mục tiêu thu được kết quả là vùng kí tự ở mức từ ta thiết lập các tham số text\_threshold = 0.7 (ngưỡng xác định đó là văn bản), link\_threshold = 0.4 (ngưỡng xác định hai kí tự kết nối với nhau) để cho hộp giới hạn có thể bao quanh văn bản trong ảnh ở mức từ, các tham số còn lại được thiết lập như mặc định như trong mã nguồn của nhóm tác giả [53]. Sau đó thực hiện các bước hậu xử lý như phân ngưỡng, lọc nhiễu… như trong [53] để trả về kết quả là một danh sách các hộp giới hạn hình chữ nhật bao quanh mỗi từ. Để thu được các từ hợp lệ ta thực hiện các bước lọc như loại bỏ các hộp giới hạn với góc nghiêng lớn, cắt đôi những hộp giới hạn có chiều cao lớn hơn ngưỡng đã thiết lập tùy theo loại chứng minh thư nhân dân, những trường hợp này xảy ra do có hai từ trên dưới dính vào nhau dẫn tới mô hình dự đoán đó là một từ. Các hộp giới hạn được tính toán lại để tạo thành các hộp giới hạn chữ nhật nằm ngang bằng cách xác định tọa độ của điểm trên cùng bên trái và điểm dưới cùng bên phải là giá trị nhỏ nhất và giá trị lớn nhất theo trục Ox Oy tương ứng. Các hộp giới hạn sau đó được cắt thành các hình ảnh của từ để phục vụ cho việc huấn luyện mô hình nhận dạng kí tự, mỗi hộp giới hạn trước khi cắt đều được mở rộng theo trục Oy 1 tỉ lệ để có thể bao phủ được các trường hợp hộp giới hạn không thể chứa được các dấu của từ.

### **3.2. Nhận dạng trích xuất thông tin**

Quá trình nhận dạng nhận dạng trích xuất thông tin chứng minh thư nhân dân được thực hiện như mô hình đã xây dựng ở Chương 2. Tuy nhiên, với bài toán đã đặt ra ở đây không chỉ yêu cầu thuật toán có độ chính xác cao mà còn phải đảm bảo tính thời gian thực. Dựa trên các kết quả nghiên cứu thực nghiệm thông qua việc khảo sát, đánh giá các hướng tiếp cận tốt hiện có, đồ án đề xuất một giải pháp phù hợp, đạt hiệu quả cao (cho độ chính xác cao và xử lý thời gian thực) để giải quyết bài toán đã đặt ra.



Hình 3.4: Quy trình nhận dạng chuyển văn bản thành giọng nói của hệ thống

Các bước cơ bản để nhận diện văn bản tiếng Việt và chuyển thành giọng nói được mô tả như trên Hình 3.4. Cũng giống cơ bản với phần Huấn luyện mô hình nhưng được chia thành 2 pha tương ứng với từng phần phát hiện vùng kí tự, nhận dạng kí tự đã được đề cập. Cả 2 pha này sẽ được thực hiện trực tiếp, liên tục và thời gian thực, pha này nối tiếp pha kia.

Tại pha 1, hình ảnh trang sách sẽ được chụp và gửi lên hệ thống thông qua máy tính kết nối mạng. Hình ảnh sẽ được truyền vào bộ phận nhận diện ký tự của hệ thống để thu được toạ độ hộp giới hạn của các từ và sắp xếp lại theo thứ tự đúng theo từng dòng.

Tại pha 2, cũng tương tự như phần xác định hình ảnh các từ trong pha 2 ở phần Huấn luyện mô hình, hình ảnh gốc và thứ tự của các từ được xử lý để thu được những ảnh của từng từ, sau đó đưa vào bộ nhận dạng ký tự (TBRA) để dự đoán. Kết quả thu được sẽ được gửi trả lại giao diện và phát cho người nghe thông qua API của Google.

## **4. Thử nghiệm hệ thống**

Quy trình thử nghiệm hệ thống bao gồm hai quy trình: huấn luyện các mô hình con của mô hình nhận dạng ký tự và thiết lập thử nghiệm phần nhận dạng của hệ thống. Chi tiết của mỗi quy trình sẽ được đề cập ở phần sau.

### **4.1. Cài đặt huấn luyện các mô hình**

Bước này bao gồm việc huấn luyện các mô hình: mô hình phân đoạn U-Net, mô hình phân loại VGG16 và mô hình nhận dạng kí tự TBRA.

Việc cài đặthuấn luyện các mô hình được thực hiện trên môi trường máy ảo cung cấp bởi Google Colaboratory. Thông số cụ thể của môi trường được mô tả cụ thể trong bảng 3.1:

|  |  |
| --- | --- |
|  | Môi trường máy ảo của Google Colaboratory |
| Vi xử lý | Intel Xeon 2.30GHz 2 luồng |
| Dung lượng RAM | 13GB |
| Dung lượng bộ nhớ | 49 GB |
| GPU | Testla K80 12GB VRAM |
| Hệ điều hành | Ubuntu 18.04 |

Bảng 3. 1: Môi trường cài đặt thử nghiệm việc huấn luyện các mô hình

Google Colaboratory là một môi trường Jupyter notebook miễn phí không yêu cầu cài đặt và hoạt động toàn bộ trên cloud. Jupyter Notebook là một ứng dụng chạy trên web cho phép người dùng tạo ra các văn bản, thường được gọi là notebook, bao gồm các đoạn code và đầu ra khi chạy chúng (có thể là chữ, hình ảnh, video, biểu đồ, v.v…). Jupyter Notebook hỗ trợ nhiều ngôn ngữ, bao gồm Python, R, Julia,… Google Colaboratory được xây dựng dựa trên Jupyter Notebook, tuy nhiên có điểm khác là cung cấp tài nguyên tính toán mạnh bao gồm: dung lượng RAM và bộ nhớ lớn, vi xử lý có hiệu năng tốt và GPU hỗ trợ tính toán. Tuy nhiên Google Colab hiện đang chỉ hỗ trợ ngôn ngữ Python. Các tài liệu chia sẻ trên Google Colaboratory đều là một notebook, nhưng được chạy trên cloud của Google Colaboratory và lưu trữ trong Google Drive tương ứng với tài khoản Google được sử dụng để tạo notebook. Môi trường của Colabotary có thể liên kết với Google Drive của người dùng để đọc và ghi các file từ đó. Môi trường có sẵn một số thư viện Python, bao gồm cà các thư viện dùng cho học máy phổ biến như TensorFlow hay scikỉt-learn.

***\* Mô hình nhận dạng kí tự TBRA***

\* Chuẩn bị dữ liệu

Như đã trình bày, đồ án huấn luyện mô hình TBRA sử dụng hai tập dữ liệu gán nhãn và tập dữ liệu sinh. Tập dữ liệu gán nhãn là những hình ảnh của các từ được cắt ra từ bước trước sau đó áp dụng mô hình có hỗ trợ sẵn tiếng Việt là Tesseract [56] để nhận dạng ký tự.

Tổng cộng lượng dữ liệu gán nhãn là 2 triệu hình ảnh bao gồm cả thực hiện tăng số lượng dữ liệu (Data Augmentation) bằng việc thêm các ảnh được làm mờ, thêm nhiễu, làm nghiêng,… Tập dữ liệu sinh là tập dữ liệu được tạo ra từ các công cụ sinh kí tự trên nền hình ảnh cho trước với các thông số về phông chữ, cỡ chữ, nền ảnh, độ nghiêng, độ nhiễu,… tùy chọn, sử dụng công cụ TextRecognitionDataGenerator [57]. Đồ án thực hiện tạo ra các từ văn bản ngẫu nhiên theo phông chữ, hình nền tương ứng với nhiều loại nền sách khác nhau. Dữ liệu cũng được làm mờ, thêm nhiễu, làm nghiêng để tăng sự đa dạng, tập dữ liệu sinh cũng gồm 2 triệu hình ảnh.

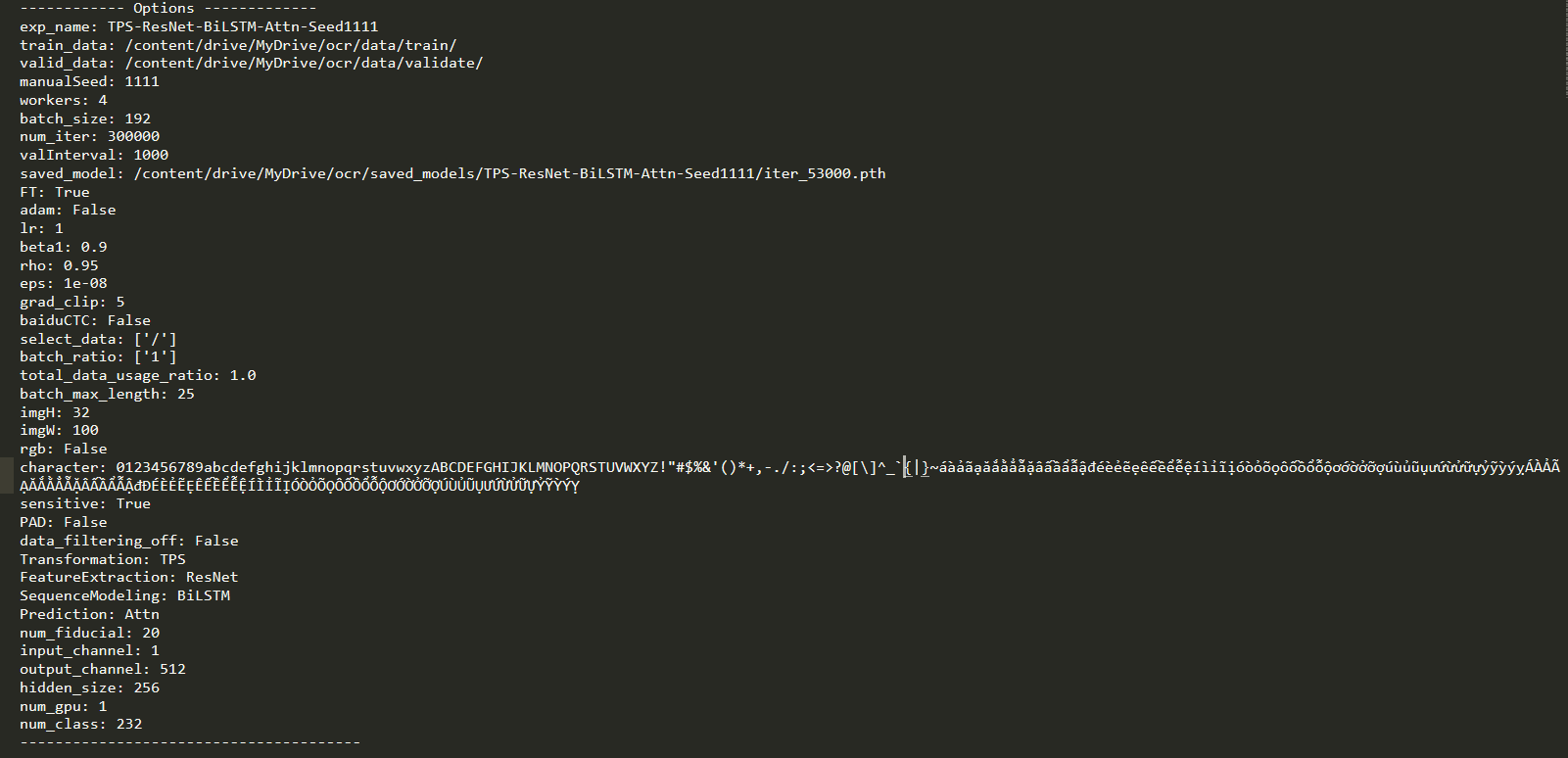
Với tập dữ liệu chia thành 2 tập con cho việc huấn luyện với tỉ lệ 80:20 trong đó 80% dữ liệu huấn luyện (training dataset) và 20% dữ liệu kiểm chứng (validation dataset). Toàn bộ dữ liệu được lưu trữ dưới dạng LMDB [55].

LMDB (Lightning Memory-Mapped Database), được phát triển và duy trì bởi Symas Corporation để thay thế [Berkeley DB](https://dbdb.io/db/berkeley-db) [52] trong dự án OpenLDAP, là một cơ sở dữ liệu nhúng có cấu trúc dạng khóa – giá trị và dựa trên cây B+. Các tính năng chính của LMDB là nó sử dụng một kho lưu trữ đơn cấp dựa trên các tệp bản đồ bộ nhớ, có nghĩa là hệ điều hành chịu trách nhiệm quản lý các trang (như bộ nhớ đệm thường xuyên sử dụng các trang). Nó sử dụng phương pháp lưu trữ copy-on-write với một luồng ghi duy nhất; những luồng đọc không chặn luồng ghi và ngược lại. Hệ thống chỉ duy trì nhiều nhất hai phiên bản dữ liệu vào bất kỳ lúc nào (tức là khi đã cam kết tất cả các phiên bản trước đó sẽ bị loại bỏ). Nó cũng duy trì một danh sách các trang trống để theo dõi và sử dụng lại các trang thay vì phân bổ bộ nhớ mỗi lần.

\* Thực hiện huấn luyện

Việc huấn luyện được thực hiện với thư viện PyTorch, được phát triển bởi Facebook và là nền tảng tính toán khoa học được xây dựng để hỗ trợ chính cho các giải thuật học máy thường sử dụng GPU. Để được xem 1 thư viện Deep Learning lý tưởng, nó phải có tính dễ học và dễ sử dụng, đủ linh hoạt để sử dụng trong các ứng dụng khác nhau, hiệu quả để chúng ta có thể xử lý các bộ dữ liệu thực tế lớn và đủ chính xác để cung cấp kết quả chính xác ngay cả khi không chắc chắn về dữ liệu đầu vào. PyTorch được đánh giá là có thể đáp ứng rất tốt các yêu cầu nêu trên. Bên cạnh đó, PyTorch còn sỡ hữu khả năng tính toán với GPU, mang lại hiệu năng tính toán cao. Ngoài ra, phần lõi của PyTorch (libtorch) được viết bằng C++ giúp cài đặt các mô hình huấn luyện dễ dàng hơn khi phải thực hiện trên nhiều môi trường khác nhau.

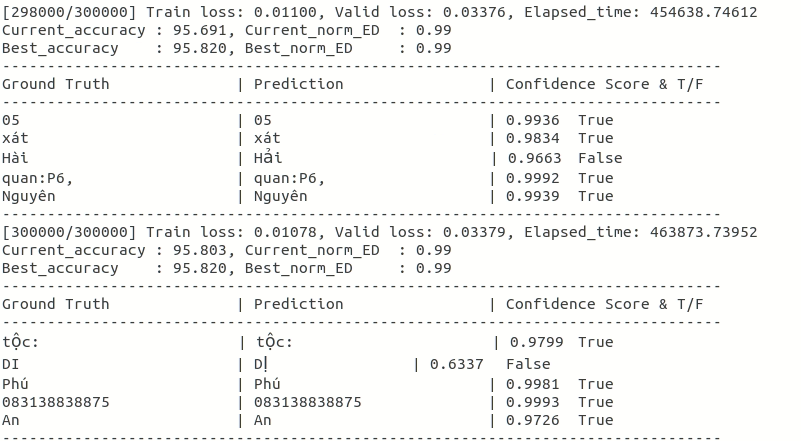
Các đoạn mã huấn luyện được đồ án lấy từ mã nguồn của nhóm tác giả [54]. Theo đó, các tham số được thiết lập cho việc huấn luyện như trên hình 3.13.



Hình 3.5: Tham số đào tạo TBRA

Trong đó select\_data, lựa chọn tập dữ liệu cho việc huấn luyện, thiết lập chọn toàn bộ tập dữ liệu huấn luyện. Batch\_ratio, tỉ lệ dữ liệu được chọn trong mỗi mẻ, được thiết lập là 1. Character, các nhãn kết quả của mỗi kí tự, được thiết lập là tất cả các kí tự trong tiếng Việt. Hình ảnh đầu vào là ảnh xám có kích thước 32x100 và batch\_size là 192. Mô hình dựa trên attention với việc so sánh các phân phối xác xuất đầu ra nên hàm mất mát là cross\_entropy, việc tối ưu hàm mất mát được thực hiện theo mặc định với hàm Adadelta [44]. Cuối cùng đồ án thực hiện huấn luyện bằng cách tinh chỉnh (fine turning) mô hình đã được đào tạo trước của nhóm tác giả với tập dữ liệu [MJSynth (MJ)](http://www.robots.ox.ac.uk/~vgg/data/text/) [1] và [SynthText (ST)](http://www.robots.ox.ac.uk/~vgg/data/scenetext/) [2].

--chưa có dữ liệu của itter 300k



Hình 3.6: Kết quả huấn luyện TBRA

Kết quả huấn luyện được thể hiện như trên hình 3.6, với 300000 iteration và độ chính xác trên tập dữ liệu kiểm chứng đạt 98.820%.

### **4.2. Nhận dạng trích xuất thông tin**

Trong phần này, đồ án sẽ trình bày về các thiết lập thử nghiệm cho phần nhận dạng và chuyển đổi văn bản thành giọng nói, chi tiết việc cài đặt, kết quả thu được, đánh giá kết quả.

#### **4.2.1. Cài đặt các hệ thống**

Hệ thống được thiết lập trên máy tính cấu hình có cấu hình như sau:

|  |  |
| --- | --- |
| Hệ điều hành | Windows 10 |
| Vi xử lý | Intel core I5-7200U 2.50GHz 4 nhân 4 luồng |
| Dung lượng RAM | 8GB |
| Dung lượng bộ nhớ | 512 Gb |
| GPU | Intel HD Graphics 620 |

Bảng 3.2: Cấu hình hệ thống thử nghiệm

\* Cài đặt máy chủ ứng dụng

Máy chủ ứng dụng được cài đặt với ngôn ngữ Python phiên bản 3.6.

Các thư viện chính của hệ thống:

|  |  |  |
| --- | --- | --- |
| STT | Tên package | Phiên bản |
| 1 | numpy | 1.19.4 |
| 2 | tensorflow | 2.0.0 |
| 3 | Keras | 2.0.0 |
| 4 | torch | 1.5.0 |
| 6 | Flask | 1.1.2 |
| 7 | opencv-python | 3.4.2.16 |

Bảng 3.3: Các thư viện chính được sử dụng trong máy chủ hệ thống

Mỗi mô đun tiền xử lý, phát hiện vùng kí tự, nhận dạng kí tự trong máy chủ ứng dụng được xây dựng thành các RESTful API tương ứng với cổng 5000 cho máy chủ web và cổng 3010 cho máy chủ nhận diện, nhận yêu cầu từ máy chủ web và dữ liệu trao đổi có dạng JSON.

\* Cài đặt máy chủ web và trình duyệt phía người dùng

|  |  |
| --- | --- |
|  | Công nghệ, ứng dụng hỗ trợ |
| Frontend | - Thiết kế giao diện, xử lý dữ liệu, logic hiển thị: thư viện ReactJs, JavaScript.  - Tạo luồng stream: Socketio |
| Backend | - Ngôn ngữ: Python.  - Sử dụng framework Flask.  - Triển khai trên local host, cổng 5000 và 3010.  - Xây dựng các RESTful API nhận hình ảnh và các thông tin liên quan từ trình duyệt người gọi máy chủ ứng dụng để thực thi nhận dạng trích xuất thông tin. Bao gồm hai máy chủ phát hiện và nhận dạng. |

Bảng 3.4: Các công nghệ sử dụng trong máy chủ web và trình duyệt phía người dùng

#### **4.2.2. Thiết lập thử nghiệm**

Mục đính chính của thử nghiệm bước nhận dạng trích xuất thông tin của hệ thống là để kiểm tra tính chính xác của hệ thống, bằng cách trả lời các câu hỏi, với một hình ảnh có chứa chứng minh thư đầu vào liệu hệ thống:

* Có thể phát hiện và sắp xếp đúng thứ tự các từ có trong ảnh hay không?
* Có thể nhận dạng đúng phần kí tự được cắt từ phần phát hiện hay không và việc hậu xử lý của phần này có tốt hay không?
* Có đáp ứng với thời gian cho phép trong việc xử lý một hình ảnh văn bản bất kỳ hay không?

Để thử nghiệm và đánh giá hiệu năng của hệ thống, đồ án thử nghiệm hệ thống với những hình ảnh chứng minh thư nhân dân không chỉ được chụp bởi thiết bị camera của hệ thống mà còn thử nghiệm với những hình ảnh khác được chụp ngẫu nhiên với các nền khác nhau và góc chụp và nền khác nhau, tổng số hình ảnh thử nghiệm là 100.

Với mỗi thực nghiệm, đồ án thống kê 3 loại kết quả:

a. Nhận dạng đúng, cụ thể:

a1. Phát hiện đụng ký tự từ ảnh và nhận diện được là từ tiếng Việt nào.

a2. Chấp nhận các kí tự nhận dạng sai khi mắt thường cũng không đọc được.

b. Không nhận dạng được, cụ thể:

b1. Không phát hiện được ký tự từ ảnh.

b2. Không nhận diện được là từ tiếng Việt nào.

c. Nhận dạng nhầm, cụ thể:

c1. Phát hiện sai vùng kí tự hoặc không phát hiện được các vùng kí tự dẫn đến kết quả không chính xác.

c2. Phát hiện được các vùng kí tự nhưng phần hậu xử lý của phát hiện vùng kí tự chưa chính xác dẫn đến thông tin kết quả sai.

c3. Phát hiện được các vùng kí tự và hậu xử lý đúng, phần nhận dạng kí tự và hậu xử lý sai.

c4. Phát hiện được các vùng kí tự và hậu xử lý đúng, phần nhận dạng kí tự đúng nhưng hậu xử lý sai.

Trong tất cả các kết quả trên, trường hợp c3 và c4 là xấu nhất khi các từ nhận dạng đúng nhưng việc hậu xử lý sai dẫn đến thông tin bị sai. Trường hợp c3 do việc hậu xử lý tạo dòng các vùng kí tự và sưaps xếp thứ từ các từ chưa chính xác, nguyên nhân thường xảy ra do các kí tự của dòng trên và dưới quá sát nhau dẫn đến việc xác định hai dòng đó là một hoặc một hộp giới hạn tuy của một từ nhưng lại to vượt ngưỡng quy định nên hộp đó sẽ bị cắt đôi theo chiều ngang dẫn tới kết quả nhận dạng không chính xác, giải pháp cho những việc trên là xây dựng thuật toán hậu xử lý vùng kí tự tốt hơn bao phủ nhiều trường hợp hơn. Các trường hợp b1, b2, c1, c2 do mô hình được huấn luyện chưa tốt, giải pháp là huấn luyện lại các mô hình với tập dữ liệu lớn hơn hoặc thay đổi các tham số huấn luyện cho phù hợp hơn.

Để đánh giá chất lượng của hệ thống, các độ đo sau đây được sử dụng: độ chính xác (precision), độ bao phủ (recall), và độ đo F1.

Với số ảnh có kết quả trích xuất thông tin là trường hợp ảnh đưa vào hệ thống có kết quả trích xuất thông tin trả ra.

#### **4.2.3. Đánh giá kết quả**

Kết quả thử nghiệm với 100 hình ảnh trang sách ngẫu nhiên như sau:

|  |  |
| --- | --- |
| Các trường hợp | Kết quả |
| Nội dung đúng, không tính trường hợp thông tin mờ | 68 |
| Phát hiện sai vùng kí tự dẫn đến kết quả sai | 11 |
| Hậu xử lý vùng kí tự dẫn đến kết quả sai | 0 |
| Nhận dạng kí tự sai dẫn đến kết quả sai | 2 |
| Nhận dạng kí tự đúng nhưng hậu xử lý sai dẫn đến kết quả sai | 1 |
| Không phát hiện được vùng kí tư | 0 |

Bảng 3.5: Kết quả thử nghiệm hệ thống

Từ kết quả thử nghiệm trên bảng 3.5 độ chính xác đạt 73.1% và độ bao phủ của hệ thống là 68%, còn F1 là 70.5%. Nhìn chung, hệ thống trích xuất đúng thông tin các trường khá chính xác, thời gian trung bình để trích xuất một ảnh dưới 2 giây. Với 10 ảnh phân đoạn sai và ảnh không phân đoạn được cho thấy mô hình phân đoạn ảnh chưa hoạt động tốt nên cần được huấn luyện lại hoặc thay đổi các tham số như đã đề cập. Về phần phát hiện vùng kí tự và nhận dạng, có 11 trường hợp phát hiện vùng kí tự sai gồm gồm vùng của từ này đè lên từ khác hoặc xác định nền là vùng kí tự ở ảnh chứng minh thư nhân dân loại cũ như trên hình 3.16do mô hình đào tạo trước của CRAFT được đào tạo với nhiều các hình ảnh khác nhau nên khi sử dụng cho riêng ảnh chứng minh thư nhân dân thì kết quả chưa tốt nhất. Và giải pháp cần là thực hiện tinh chỉnh mô hình đào tạo trước CRAFT với tập dữ liệu chỉ có ảnh chứng minh thư nhân dân.

## **5. Kết luận**

Chương này trình bày quy trình đồ án áp dụng để thử nghiệm và đánh giá hệ thống. Do dữ liệu ít nên chỉ thử nghiệm đánh giá trên 300 bộ dữ liệu mẫu. Từ kết quả thu được, có thể rút ra kết luận hệ thống hoạt động ổn định với các trường hợp ảnh được chụp từ thiết bị trong hệ thống rõ ràng, điều kiện ánh phù hợp. Còn với các trường hợp ảnh mờ, dữ liệu trong chứng minh thư bị nhòe màu do để thời gian lâu, các trường in lộn xộn vị trí thì hệ thống có thể nhận dạng nhầm hoặc sai. Trong những trường hợp đó cần giao dịch viên kiểm tra lại kĩ càng.

# CHƯƠNG 4: KẾT LUẬN

## **1. Đóng góp của đồ án**

Với sự phát triển của các phương pháp học sâu, độ chính xác của các bài toán phân đoạn, phát hiện, nhận dạng ngày càng cao. Một mặt nhờ các giải thuật tiên tiến, một mặt nhờ

Đồ án này đã tập trung nghiên cứu công nghệ nhận dạng kí tự quang học và xây dựng hệ thống nhận dạng trích xuất thông tin chứng minh thư nhân dân áp dụng trong các phòng giao dịch ngân hàng. Kết quả đạt được của đồ án bao gồm:

- Làm chủ quy trình, các bước để xây dựng một hệ thống áp dụng công nghệ nhận dạng kí tự quang học.

- Làm chủ được giải pháp công nghệ và các kỹ thuật hiện đại để giải quyết bài toán nhận dạng kí tự quang học, cụ thể bao gồm:

+ Các phương pháp phân đoạn, phát hiện hình ảnh, phương pháp phát hiện các vùng kí tự trong hình ảnh và nhận dạng kí tự áp dụng mạng neuron học sâu.

+ Các phương pháp xử lý hình ảnh số, các giải thuật để phục vụ trong việc xử lý các chuỗi kí tự.

- Nắm được việc xây dựng cơ bản hệ thống bao gồm cả Frontend và Backend với các ngôn ngữ Javascript và Python.

- Hệ thống chuyển đổi hình ảnh văn bản thành giọng nói đạt độ chính xác cao.

## **2. Hướng phát triển tiếp theo**

Trong quá trình triển khai, khó khăn lớn nhất đồ án gặp phải đó là quá trình thu thập dữ liệu, vì số lượng ảnh chứng minh thư nhân dân lớn, dữ liệu thu thập lại thường bị ảnh hưởng do điều kiện môi trường (ánh sáng, nhiễu …). Nên trong tương lai nhóm tác giả sẽ nghiên cứu, phát triển công cụ dựa trên mạng xã hộ để thu thập dữ liệu ảnh tự động và bán tự động. Tiếp nữa đồ án muốn phát phiển hệ thống có khả năng trích xuất thông tin của nhiều loại giấy tờ hơn như bằng lái xe, hộ chiếu hay chứng minh thư nhân dân.

# TÀI LIỆU THAM KHẢO

**Tài liệu tiếng Anh:**

[1]: O. Ronneberger, P. Fischer, and T. Brox. “U-net: Convolutional networks for biomedical image segmentation”. In MICCAI, 2015.

[2]: Y. Bengio, P. Y. Simard, and P. Frasconi (1994). “Learning longterm dependencies with gradient descent is difficult”.

[3]: He T, Huang W, Qiao Y, Yao J (2016). “Text-attentional convolutional neural networks for scene text detection”. In IEEE Trans Image Process 25:2529–2541.

[4]: Babenko B, Belongie S (2012). “End-to-end scene text recognition”. In: IEEE international conference on computer vision.

[5]: Navneet Dalal and Bill Triggs (2005). “Histograms of Oriented Gradients for Human Detection”. In CVPR05.

[6]: Diederik P. Kingma, Jimmy Ba. “Adam: A Method for Stochastic Optimization”. In arXiv:1412.6980, 2014.

[7]: Epshtein B, Ofek E, Wexler Y (2010). “Detecting text in natural scenes with stroke width transform”. In: Computer vision & pattern recognition, pp 2963–2970.

[8]: Miron B. Kursa, “rFerns: An Implementation of the Random Ferns Method for General-Purpose Machine Learning”. In arXiv:1202.1121, 2014.

[9]: Pan YF, Hou X, Liu CL (2011). “A hybrid approach to detect and localize texts in natural scene images”.

[10]: Mishra A, Alahari K, Jawahar CV (2012) “Scene text recogni-tion using higher order language priors”. In: Proceedings british machine vision conference, pp 1–11.

[11]: Zhu Y, Yao C, Bai X (2016). “Scene text detection and recognition: recent advances and future trends”.

[12]: Baek, Youngmin et al. “Character Region Awareness for Text Detection.” 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019): 9357-9366.

[13]: A. Graves, A. Mohamed, and G. E. Hinton. “Speech recognition with deep recurrent neural networks”. In ICASSP, 2013.

[14]: B. Epshtein, E. Ofek, and Y. Wexler. “Detecting text in natural scenes with stroke width transform”. In CVPR, pages 2963– 2970. IEEE, 2010.

[15]: J. Matas, O. Chum, M. Urban, and T. Pajdla. “Robust widebaseline stereo from maximally stable extremal regions”. Image and Vision Computing, 22(10):761–767, 2004

[16]: Canny J. “A Computational Approach To Edge Detection”. IEEE Transactions on Pattern Analysis and Machine Intelligence, 8(6):679–698, 1986.

[17]: K. Simonyan and A. Zisserman. “Very deep convolutional networks for large-scale image recognition”. In ICLR, 2015.

[18]: Baoguang Shi, Xiang Bai, and Cong Yao. “An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition”. In TPAMI, volume 39, pages 2298–2304. IEEE, 2017

[19]: A. Krizhevsky, I. Sutskever, and G. E. Hinton. “Imagenet classification with deep convolutional neural networks”. In NIPS, 2012

[20]: R. B. Girshick, J. Donahue, T. Darrell, and J. Malik. “Rich feature hierarchies for accurate object detection and semantic segmentation”. In CVPR, 2014

[21]: M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. “Reading text in the wild with convolutional neural networks”. IJCV (Accepted), 2015

[22]: Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. “Spatial transformer networks”. In NIPS, pages 2017–2025, 2015

[23]: Chen-Yu Lee and Simon Osindero. “Recursive recurrent nets with attention modeling for ocr in the wild”. In CVPR, pages 2231–2239, 2016

[24]: A. Graves, S. Fernandez, F. J. Gomez, and J. Schmidhu-ber. “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks”. In ICML, 2006.

[25]: Baoguang Shi, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. “Robust scene text recognition with automatic rectification”. In CVPR, pages 4168–4176, 2016.

[26]: Zhanzhan Cheng, Fan Bai, Yunlu Xu, Gang Zheng, Shiliang Pu, and Shuigeng Zhou. “Focusing attention: Towards accurate text recognition in natural images”. In ICCV, pages 5086–5094, 2017.

[27]: S. Hochreiter and J. Schmidhuber. “Long short-term memory”. Neural Computation, 9(8):1735–1780, 1997.

[28]: Baek, Jeonghun et al. “What Is Wrong With Scene Text Recognition Model Comparisons? Dataset and Model Analysis.” 2019 IEEE/CVF International Conference on Computer Vision (ICCV) (2019): 4714-4722.

[29]: Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep residual learning for image recognition”. In CVPR, pages 770–778, 2016.

[30]: Wei Liu, Chaofeng Chen, Kwan-Yee K Wong, Zhizhong Su, and Junyu Han. “Star-net: A spatial attention residue network for scene text recognition”. In BMVC, volume 2, 2016.

[31]: Jianfeng Wang and Xiaolin Hu. “Gated recurrent convolution neural network for ocr”. In NIPS, pages 334–343, 2017.

[32]: Fedor Borisyuk, Albert Gordo, and Viswanath Sivakumar. “Rosetta: Large scale system for text detection and recognition in images”. In KDD, pages 71–79, 2018.

[33]: Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. “Neural machine translation by jointly learning to align and translate”. In ICLR, 2015.

[34]: K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio. “On the properties of neural machine translation: Encoder-decoder approaches”. CoRR, abs/1409.1259, 2014.

[35]: A. Graves, G. Wayne, I. Danihelka (2014). "Neural Turing Machines". In arXiv:1410.5401.

[36]: David G. Lowe, "Object recognition from local scale-invariant features," International Conference on Computer Vision, Corfu, Greece (1999).

[37]: Minh-Thang Luong, Hieu Pham, Christopher D. Manning (2015). “Effective Approaches to Attention-based Neural Machine Translation”. In arXiv:1508.04025.

[38]: Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner (1988). “Gradient-Based Learning Applied To Document Recognition”. In IEEE.

[39]: Dario Garcia-Gasulla and Ferran Parés and Armand Vilalta and Jonatan Moreno and Eduard Ayguadé and Jesús Labarta and Ulises Cortés and Toyotaro Suzumura, “On the Behavior of Convolutional Nets for Feature Extraction”. In arXiv:1703.01127.

**Danh mục các website tham khảo:**

[40]: https://cvit.iiit.ac.in/research/projects/cvit-projects/the-iiit-5k-word-dataset

[41]: https://rrc.cvc.uab.es/

[42]: https://machinelearningcoban.com/2017/01/12/gradientdescent/

[43]: https://en.wikipedia.org/wiki/Hadamard\_product\_(matrices)

[44]: https://d2l.aivivn.com/chapter\_convolutional-modern/resnet\_vn.html

[45]: https://towardsdatascience.com/everything-you-ever-wanted-to-know-about-computer-vision-heres-a-look-why-it-s-so-awesome-e8a58dfb641e

[46]: https://innotech-vn.com/giai-phap-ocr-do-chinh-xac-cao-cho-ngan-hang-tai-viet-nam/

[47]: http://csnd.vn/Home/Nghien-cuu-Trao-doi/2503/Tim-hieu-ve-cong-nghe-nhan-dang-ki-tu-quang-hoc-OCR-Optical-Character-Recognition-ung-dung-trong-cac-phan-mem-phat-hien-va-xu-ly-vi-pham-luat-giao-thong-thong-qua-he-thong-camera-giam-sat.

[48]: <https://forum.machinelearningcoban.com/t/kien-truc-cac-mang-cnn-noi-tieng-phan-1-alex-lenet-inception-vgg/2582>

[49]: <http://www.image-net.org/>

[50]: <https://www.researchgate.net/publication/330331570_Review_of_Scene_Text_Detection_and_Recognition>

[51]: https://lib.hpu.edu.vn/bitstream/handle/123456789/18326/66\_DoHongKien\_CT1201.pdf

[52]: https://www.oracle.com/database/technologies/related/berkeleydb.html

[53]: https://github.com/clovaai/CRAFT-pytorch

[54]: https://github.com/clovaai/deep-text-recognition-benchmark

[55]: https://lmdb.readthedocs.io/en/release/

[56]: https://github.com/tesseract-ocr/tesseract

[57]: https://github.com/Belval/TextRecognitionDataGenerator

[58]: https://github.com/donydchen/landmark-tool

[59]: https://github.com/fpt-corp/VN-address-correction

[60]: https://github.com/zhixuhao/unet

[61]: https://en.wikipedia.org/wiki/Levenshtein\_distance

[62]: https://nguyentruonglong.net/giai-thich-chi-tiet-ve-mang-long-short-term-memory-lstm.html

[63]: https://viblo.asia/p/machine-learning-attention-attention-attention-eW65GPJYKDO

[64]: https://towardsdatascience.com/understanding-semantic-segmentation-with-unet-6be4f42d4b47