**CÔNG VIỆC 7: XÂY DỰNG MODULE NHẬN DẠNG KÝ TỰ QUANG HỌC**

# **LỜI NÓI ĐẦU**

Nhu cầu trích xuất văn bản từ hình ảnh ngày càng gia tăng trong nhiều lĩnh vực, từ việc số hóa tài liệu giấy đến tự động hóa quy trình xử lý thông tin. Công nghệ nhận dạng ký tự quang học, hay Optical Character Recognition (OCR), đã và đang đóng vai trò quan trọng trong việc chuyển đổi hình ảnh chứa văn bản, cho dù đó là chữ viết tay, chữ in, hoặc ký tự đánh máy, thành các ký tự mà máy tính có thể nhận diện và xử lý. Với OCR, việc chuyển đổi các tài liệu từ dạng hình ảnh sang dạng văn bản số hóa trở nên dễ dàng hơn, giúp tiết kiệm thời gian và công sức.

Trong nhiều tình huống, việc chỉnh sửa nội dung của tài liệu giấy như bài viết trên tạp chí, tờ rơi quảng cáo, hoặc thậm chí là các tập tin PDF, thường gặp nhiều khó khăn bởi chúng không ở định dạng số hóa có thể dễ dàng xử lý. Các định dạng như .doc, .docx, .csv, .txt… là những định dạng mà máy tính có thể đọc và hiểu được cả văn bản lẫn cấu trúc của nó. Chính vì vậy, công nghệ OCR trở nên vô cùng cần thiết, bởi nó cho phép chúng ta chuyển đổi các tài liệu hình ảnh không có cấu trúc thành văn bản số hóa có thể chỉnh sửa, lưu trữ và tìm kiếm dễ dàng.

Tài liệu này nhằm mục đích nghiên cứu sâu hơn về các công nghệ tiên tiến đang được sử dụng trong OCR để thực hiện trích xuất văn bản từ hình ảnh một cách hiệu quả. Tài liệu tập trung vào nghiên cứu các thuật toán, mô hình máy học, và các kỹ thuật xử lý hình ảnh đằng sau OCR, cùng với những thách thức và cơ hội trong việc áp dụng công nghệ này trong các ứng dụng thực tế. Qua đó, tài liệu không chỉ cung cấp kiến thức cơ bản về OCR mà còn hướng tới việc giúp người đọc nắm bắt được xu hướng phát triển và tiềm năng ứng dụng của công nghệ này trong tương lai.

# **MỤC LỤC**

[LỜI NÓI ĐẦU 2](#_Toc176568353)

[MỤC LỤC 3](#_Toc176568354)

[DANH MỤC HÌNH ẢNH 4](#_Toc176568355)

[DANH MỤC BẢNG BIỂU 6](#_Toc176568356)

[DANH MỤC CHỮ VIẾT TẮT 7](#_Toc176568357)

[CHƯƠNG 1: GIỚI THIỆU BÀI TOÁN OCR 8](#_Toc176568358)

[1.1.Giới thiệu 8](#_Toc176568359)

[1.2.Ứng dụng của OCR 9](#_Toc176568360)

[CHƯƠNG 2: NHẬN DẠNG KÝ TỰ QUANG HỌC VỚI OCR 12](#_Toc176568361)

[2.1 Image Pre-processing 13](#_Toc176568362)

[2.2 Text Detection 15](#_Toc176568363)

[2.3. Detection Boxes Rectify 18](#_Toc176568364)

[2.4.Text Recognition 19](#_Toc176568365)

[3.1. Standard binarization 22](#_Toc176568366)

[3.2. Differentiable binarization 23](#_Toc176568367)

[3.3. Ngưỡng động (Adaptive threshold) 23](#_Toc176568368)

[3.4. Hàm mất mát 24](#_Toc176568369)

[CHƯƠNG 4: NHẬN DIỆN VĂN BẢN VỚI MÔ HÌNH CRNN 26](#_Toc176568370)

[4.1 Convolutional Layers 27](#_Toc176568371)

[4.2 Recurrent Layers 28](#_Toc176568372)

[4.3 Transcription Layers 29](#_Toc176568373)

[4.4. Connectionist Temporal Classification 31](#_Toc176568374)

[CHƯƠNG 5: THỬ NGHIỆM VÀ ĐÁNH GIÁ 33](#_Toc176568375)

[KẾT LUẬN 35](#_Toc176568376)

[TÀI LIỆU THAM KHẢO 36](#_Toc176568377)

# **DANH MỤC HÌNH ẢNH**

[Hình 1.Chuyển đổi hình ảnh và tải liệu thành văn bản số hóa 7](#_Toc176568276)

[Hình 2.Ứng dụng của OCR trong nhập liệu tự động 7](#_Toc176568277)

[Hình 3.Ứng dụng của OCR trong nhận dạng biển số xe 8](#_Toc176568278)

[Hình 4.Ứng dụng của OCR trong nhận dạng biển báo giao thông 8](#_Toc176568279)

[Hình 5.Ứng dụng của OCR trong quét tài liệu 9](#_Toc176568280)

[Hình 6.Các module chính trong hệ thống OCR 10](#_Toc176568281)

[Hình 7.Kết quả sau khi tiền xử lý hình ảnh 12](#_Toc176568282)

[Hình 8.Kết quả module Text Detection 13](#_Toc176568283)

[Hình 9.Ví dụ về văn bản có cấu trúc 14](#_Toc176568284)

[Hình 10.Ba kiểu phát hiện văn bản trong OCR 14](#_Toc176568285)

[Hình 11.Biến đổi affine để thực hiện căn chỉnh góc hình ảnh 17](#_Toc176568286)

[Hình 12.Kết quả module Text Recognition 17](#_Toc176568287)

[Hình 13.Kiến trúc mô hình DB 20](#_Toc176568288)

[Hình 14.Bản đồ ngưỡng động 22](#_Toc176568289)

[Hình 15.Kiến trúc mô hình CRNN 24](#_Toc176568290)

[Hình 16.Kết quả Convolutional Layers của ảnh đầu vào 26](#_Toc176568291)

[Hình 17.Feature sequence được đưa qua Recurrent Layers 26](#_Toc176568292)

[Hình 18.Kết quả Transcription Layers 27](#_Toc176568293)

[Hình 19.Đánh nhãn cho mỗi Timestamp 28](#_Toc176568294)

[Hình 20.Kết quả của X sau khi đưa qua CTC 29](#_Toc176568295)

# **DANH MỤC BẢNG BIỂU**

[Bảng 1.Chi tiết về tập dữ liệu đào tạo và kiểm tra 33](#_Toc176568318)

[Bảng 2.Chi tiết về các tham số huấn luyện 33](#_Toc176568319)

[Bảng 3.So sánh hiệu suất của 5 kiến trúc khác nhau trong mô đun nhận diện văn bản 34](#_Toc176568320)

[Bảng 4.Hiệu suất của mô hình phát hiện văn bản khi sử dụng FPGM 34](#_Toc176568321)

[Bảng 5.So sánh hiệu suất của 5 kiến trúc khác nhau trong mô đun nhận diện văn bản 34](#_Toc176568322)

[Bảng 6.Hiệu suất của mô hình nhận diện văn bản khi sử dụng PACT 34](#_Toc176568323)

# **DANH MỤC CHỮ VIẾT TẮT**

|  |  |  |
| --- | --- | --- |
| **STT** | **Ký hiệu chữ viết tắt** | **Chữ viết đầy đủ** |
| 1 | BCE | Binary Cross-Entropy |
| 2 | CNN | Convolutional Neural Networks |
| 3 | CRAFT | Character Region Awareness for Text Detection |
| 4 | CRNN | Convolutional Recurrent Neural Network |
| 5 | CTC | Connectionist Temporal Classification |
| 6 | CTPN | Connectionist Text Proposal Network |
| 7 | DB | Differentiable Binarization |
| 8 | EAST | Efficient and Accurate Scene Text Detector |
| 9 | FOTS | Fast Oriented Text Spotting |
| 10 | GT | Ground Truth |
| 11 | KNN | K-Nearest Neighbors |
| 12 | MSER | Maximally Stable Extremal Regions |
| 13 | OCR | Optical Character Recognition |
| 14 | RNN | Recurrent Neural Network |
| 15 | SVM | Support Vector Machines |
| 16 | SWT | Stroke Width Transform |
| 17 | FPGM | Filter Pruning and Growth Model |
| 18 | PACT | Policy-aware Activation Clipping and Thresholding |

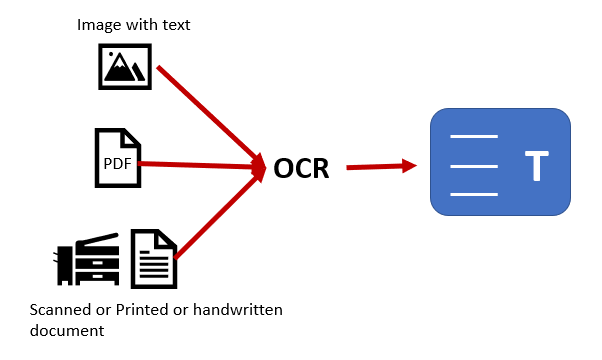
# 

# **CHƯƠNG 1: GIỚI THIỆU BÀI TOÁN OCR**

## **1.1.Giới thiệu**

Công nghệ Nhận dạng Ký tự Quang học, hay còn được gọi tắt là OCR, là một bước đột phá quan trọng trong lĩnh vực xử lý hình ảnh và văn bản. OCR cho phép máy tính chuyển đổi các văn bản và ký tự xuất hiện trong hình ảnh, tài liệu quét (scan), hoặc các định dạng phi cấu trúc khác thành dạng văn bản mà máy tính có thể hiểu và xử lý. Đây là một công nghệ giúp biến đổi các nội dung mà trước đây chỉ có thể đọc được bằng mắt thành dữ liệu số hóa có thể dễ dàng chỉnh sửa, tìm kiếm, lưu trữ và phân tích.

Trong các hệ thống máy tính truyền thống, khi ta cung cấp một hình ảnh chứa văn bản hoặc ký tự, máy tính sẽ chỉ coi đó là một tập hợp các điểm ảnh (pixels) với giá trị màu sắc và cường độ sáng khác nhau. Với dữ liệu như vậy, máy tính chỉ có thể xử lý hình ảnh ở cấp độ hình ảnh, không hiểu được nội dung văn bản trong đó. Điều này có nghĩa là việc chỉnh sửa hay tìm kiếm thông tin trong hình ảnh là không thể, hoặc rất khó khăn và tốn kém thời gian nếu thực hiện thủ công.



Hình 1.Chuyển đổi hình ảnh và tải liệu thành văn bản số hóa

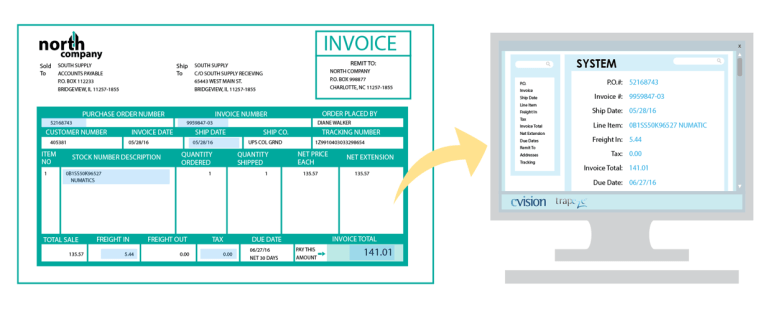
Tuy nhiên, với sự ra đời của công nghệ OCR, chúng ta có thể vượt qua những hạn chế này. OCR phân tích cấu trúc của hình ảnh, nhận diện và giải mã các ký tự, chữ viết, từ đó chuyển đổi chúng thành văn bản số hóa. Quá trình này bao gồm nhiều bước phức tạp như tiền xử lý hình ảnh, nhận dạng ký tự, và hậu xử lý để đảm bảo độ chính xác cao nhất cho văn bản đầu ra. Kết quả là, những tài liệu giấy, hình ảnh chứa văn bản như hóa đơn, sách, tài liệu lịch sử, hay thậm chí là biển số xe có thể được chuyển đổi thành các tệp văn bản số, giúp việc lưu trữ, tìm kiếm, và chỉnh sửa trở nên dễ dàng hơn rất nhiều.

## **1.2.Ứng dụng của OCR**

OCR có rất nhiều ứng dụng trong thực tế. Có thể kể đến như:

-Automatic Data Entry-Tự động nhập liệu

Đây là ứng dụng phổ biến nhất và quan trọng nhất của OCR. Trước đây, đối với những số liệu trong một hình ảnh hay tài liệu scan, để đưa vào máy tính xử lý, con người phải nhập thủ công bằng tay. Việc này rất mất thời gian và nhàm chán. Ngày nay, với sự hỗ trợ của OCR, quá trình này diễn ra hoàn toàn tự động, nhanh chóng, dễ dàng, độ chính xác cao.



Hình 2.Ứng dụng của OCR trong nhập liệu tự động

-Nhận diện biển số xe

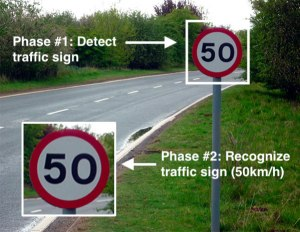
Áp dụng trong các bãi đỗ xe, tự động nhận diện biển số giúp giảm thời gian quản lý cho cả người lái xe và nhân công bảo vệ.



Hình 3.Ứng dụng của OCR trong nhận dạng biển số xe

**-Xe tự lái**

OCR giúp xe tự động nhận diện biển số để đi theo đúng chỉ dẫn.



Hình 4.Ứng dụng của OCR trong nhận dạng biển báo giao thông

-Booking Scanning

Một trong những ứng dụng phổ biến nhất của OCR là trong các máy quét tài liệu. Khi được tích hợp vào máy quét, OCR không chỉ cho phép máy quét số hóa hình ảnh của tài liệu mà còn chuyển đổi văn bản trong hình ảnh đó thành văn bản có thể chỉnh sửa và tìm kiếm. Điều này đặc biệt hữu ích trong các văn phòng, nơi mà tài liệu giấy cần được lưu trữ dưới dạng số hóa để dễ dàng quản lý và truy xuất.

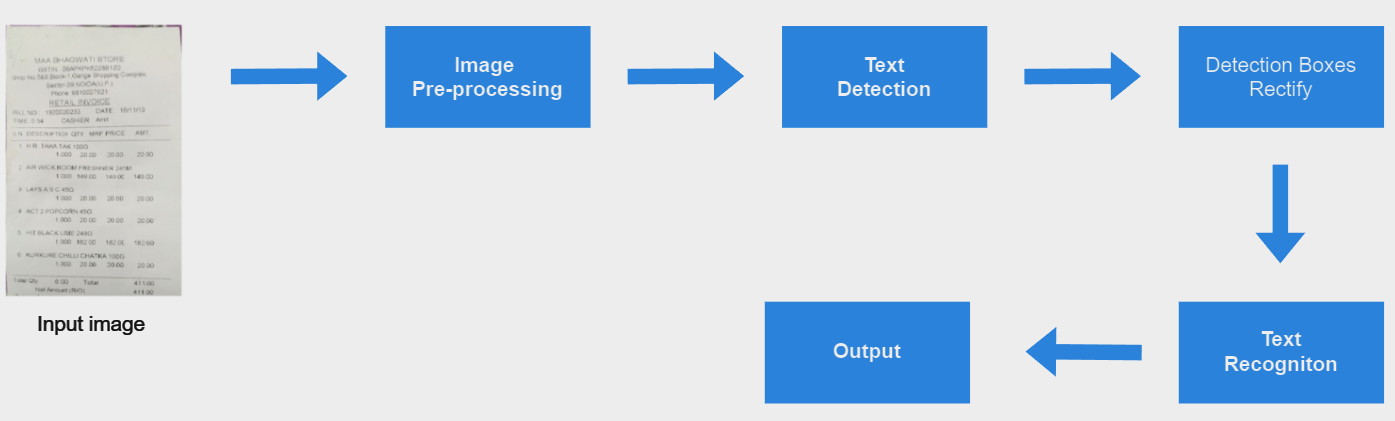


Hình 5.Ứng dụng của OCR trong quét tài liệu

OCR còn đóng vai trò quan trọng trong việc cải thiện cuộc sống của người cao tuổi và người khiếm thị. Các phần mềm OCR có thể nhận dạng văn bản và chuyển đổi nó thành giọng nói thông qua công nghệ tổng hợp giọng nói. Điều này giúp những người có thị lực kém hoặc không thể đọc được văn bản trên giấy có thể tiếp cận thông tin một cách dễ dàng và thuận tiện hơn.Và còn rất nhiều ứng dụng khác nữa…

# **CHƯƠNG 2: NHẬN DẠNG KÝ TỰ QUANG HỌC VỚI OCR**

OCR hoạt động theo chu trình như sau:



Hình 6.Các module chính trong hệ thống OCR

Đây là một hệ thống OCR 2 module bao gồm module phát hiện văn bản(Text Detection) và module nhận dạng văn bản(Text Recognition). Đây là những thành phần cốt lõi giúp hệ thống có khả năng nhận diện và trích xuất văn bản từ hình ảnh một cách hiệu quả và chính xác.

Đầu tiên, Module Phát hiện Văn bản đảm nhiệm vai trò quan trọng trong việc xác định các khu vực có chứa văn bản trong hình ảnh. Khi một hình ảnh được đưa vào hệ thống, module này sẽ phân tích và tìm kiếm những vùng có khả năng cao là chứa văn bản, sau đó tạo ra các hộp giới hạn (bounding boxes) để khoanh vùng các đoạn văn bản đó. Bằng cách này, hệ thống có thể tập trung xử lý chính xác các vùng chứa thông tin hữu ích mà không bị phân tâm bởi những vùng khác của hình ảnh.

Tiếp theo, Module Nhận dạng Văn bản sẽ tiếp nhận các vùng văn bản đã được xác định và thực hiện quá trình nhận dạng để chuyển đổi chúng thành các ký tự và từ ngữ mà máy tính có thể hiểu và xử lý. Đây là bước quan trọng để biến các hình ảnh chứa văn bản thành các dữ liệu số hóa, có thể dễ dàng tìm kiếm, chỉnh sửa, và lưu trữ. Tuy nhiên, việc nhận dạng văn bản không phải lúc nào cũng đơn giản, đặc biệt là khi văn bản trong hình ảnh bị xoay, nghiêng, hoặc không nằm ở vị trí chuẩn mực.

Để khắc phục những thách thức này, một module được tích hợp thêm là Detection Boxes Rectify - Bộ chỉnh hướng hộp phát hiện. Module này được thiết kế để giải quyết vấn đề về hướng của văn bản, đảm bảo rằng ngay cả khi văn bản bị xoay hoặc nằm ở góc độ không thuận lợi, hệ thống vẫn có thể nhận diện chính xác. Bộ chỉnh hướng này hoạt động như một bộ phân loại hướng, tự động phát hiện và điều chỉnh góc độ của các hộp phát hiện trước khi chúng được đưa vào quá trình nhận dạng. Ngoài ra, trước khi đưa hình ảnh vào mô hình, hình ảnh được đưa qua module Image Processing, đây là giai đoạn tiền xử lý nơi mà các hình ảnh đầu vào có thể gặp phải nhiều vấn đề như mở, nhiễu, lệch hướng hay các lỗi khác có thể ảnh hưởng đến chất lượng nhận dạng. Nếu những hình ảnh này được đưa trực tiếp vào hệ thống mà không qua bất kỳ xử lý nào, khả năng cao kết quả đầu ra sẽ bị ảnh hưởng, dẫn đến việc nhận dạng không chính xác hoặc thiếu sót.

## **2.1 Image Pre-processing**

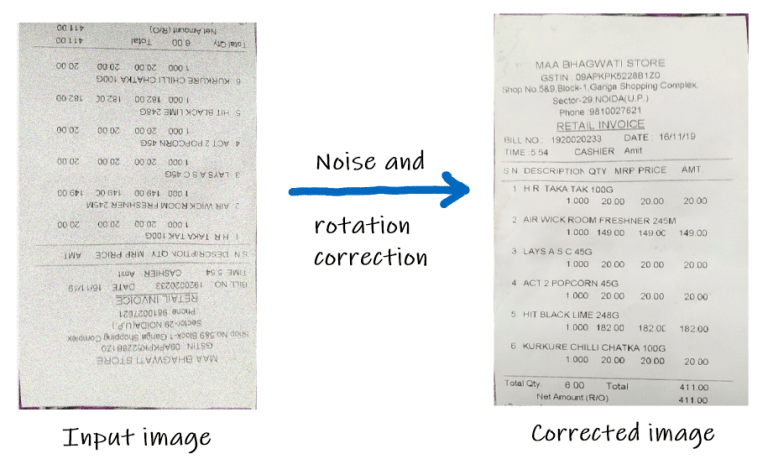
Khi làm việc với hình ảnh từ các nguồn khác nhau, chất lượng của chúng có thể bị ảnh hưởng bởi nhiều yếu tố như độ mờ, nhiễu, hay lệch hướng. Những yếu tố này nếu không được xử lý sẽ làm giảm đáng kể hiệu suất của mô hình nhận dạng, dẫn đến kết quả không chính xác. Chính vì vậy, nhiệm vụ của Image Pre-processing là cải thiện chất lượng hình ảnh, loại bỏ hoặc giảm thiểu các lỗi có thể gây trở ngại cho quá trình nhận dạng văn bản. Cụ thể, quá trình tiền xử lý hình ảnh bao gồm các phép biến đổi và kỹ thuật sau:

-Làm sắc nét hình ảnh: Khi hình ảnh bị mờ, các chi tiết của văn bản trở nên khó nhận diện hơn. Kỹ thuật làm sắc nét (sharpening) được áp dụng để tăng cường các đường nét và cạnh trong hình ảnh, giúp văn bản trở nên rõ ràng hơn

-Lọc nhiễu (Noise Reduction): Hình ảnh có thể chứa nhiều nhiễu (noise), đặc biệt là khi được chụp từ môi trường thiếu sáng hoặc từ các tài liệu cũ. Nhiễu có thể làm sai lệch thông tin văn bản, do đó, các kỹ thuật lọc nhiễu như Gaussian Blur hoặc Median Filter được sử dụng để loại bỏ những điểm ảnh không mong muốn, giúp văn bản trở nên dễ đọc hơn

-Điều chỉnh độ sáng và tương phản: Độ sáng quá cao hoặc quá thấp, cũng như tương phản kém, có thể làm giảm khả năng nhận diện của hệ thống. Các kỹ thuật điều chỉnh độ sáng (brightness) và tương phản (contrast) sẽ đảm bảo rằng các ký tự trong hình ảnh được thể hiện rõ ràng, không bị lẫn vào nền hoặc các vùng tối.

-Căn chỉnh hình ảnh (Image Alignment): Đôi khi, hình ảnh có thể bị lệch hướng, làm cho văn bản không nằm ngang hoặc không thẳng hàng. Việc căn chỉnh lại hình ảnh giúp đưa các ký tự về vị trí chuẩn, đảm bảo rằng chúng được đọc một cách chính xác bởi hệ thống. Các phép biến đổi hình học như xoay, dịch chuyển, hoặc thay đổi tỷ lệ có thể được áp dụng để đạt được điều này.

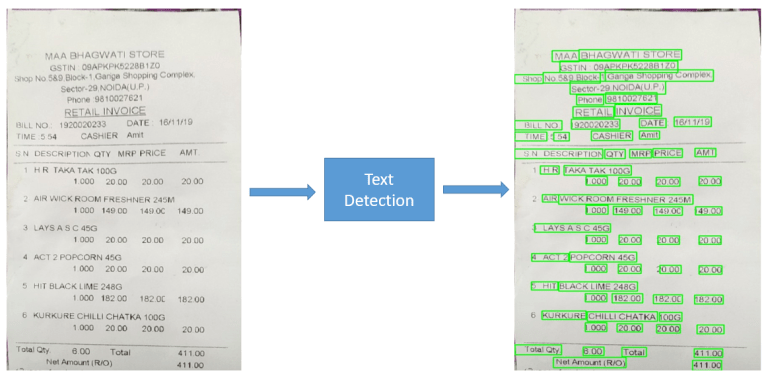


Hình 7.Kết quả sau khi tiền xử lý hình ảnh

Thông qua các kỹ thuật trên, Image Pre-processing đóng vai trò như một lớp bảo vệ, đảm bảo rằng hình ảnh đầu vào đạt chất lượng tối ưu trước khi chúng được đưa vào module học tập của hệ thống OCR. Điều này không chỉ giúp tăng cường độ chính xác của quá trình nhận dạng mà còn cải thiện hiệu suất tổng thể của hệ thống, đảm bảo rằng kết quả đầu ra là chính xác và tin cậy nhất.

## **2.2 Text Detection**

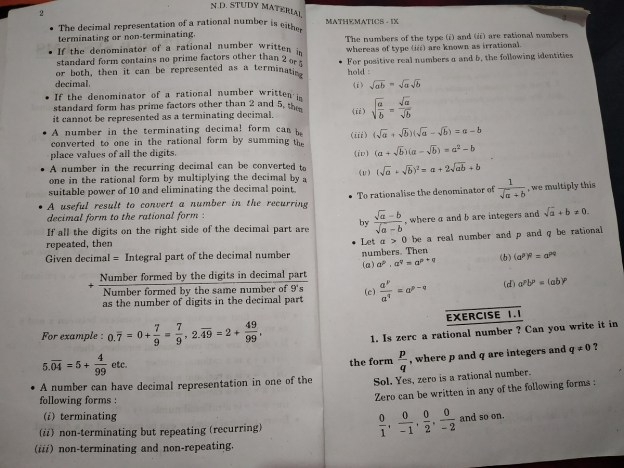
Phát hiện chữ trong hình ảnh( Text Detection) là quá trình xác định vị trí của các vùng chứa văn bản trong một hình ảnh. Mục tiêu chính của bài toán này là tìm ra các vùng trong ảnh có khả năng chứa ký tự hoặc văn bản, để từ đó thực hiện các bước xử lý tiếp theo như nhận diện văn bản( Text Recognition).



Hình 8.Kết quả module Text Detection

Có 2 kiểu ký tự/văn bản trong hình ảnh mà bài toán OCR có thể giải quyết:

-Ký tự/văn bản có cấu trúc: là những hình ảnh tương đối rõ ràng, background cố định, font chữ ổn định, ký tự/văn bản được tổ chức ngay ngắn theo hàng/cột…Ví dụ, trang sách.



Hình 9.Ví dụ về văn bản có cấu trúc

-Ký tự/văn bản phi cấu trúc: Là những hình ảnh có ký tự/văn bản xuất hiện không thống nhất về màu sắc, vị trí, kiểu chữ,... Ví dụ: bảng quảng cáo.

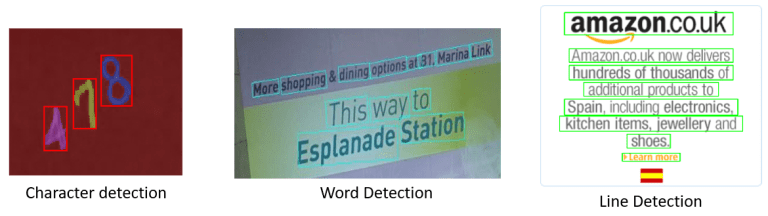
Rõ ràng, giải quyết kiểu thứ 2 khó hơn rất nhiều so với kiểu thứ nhất.

Để thực hiện nhiệm vụ Text Detection, có thể tiếp cận theo 3 cách:

-Cách 1: Phát hiện từng ký tự một( Character-by-Character)

-Cách 2: Phát hiện từng từ một( Word-by-Word)

-Cách 3: Phát hiện từng dòng một( Line-by-Line)



Hình 10.Ba kiểu phát hiện văn bản trong OCR

Nhìn chung, hầu hết các hệ thống OCR đều sử dụng cách tiếp cận thứ 2 hoặc 3. Cách 1 chậm và độ chính xác thấp hơn.

Về mặt kỹ thuật, có 2 cách có thể sử dụng:

-Cách 1: Sử dụng các kỹ thuật xử lý ảnh cơ bản(truyền thống). Cách này sử dụng các bộ lọc(filters) để tách rời các ký tự ra khỏi nền của bức ảnh, sau đó áp dụng các kỹ thuật phát hiện biên, Contours để thu được vị trí của từng ký tự riêng rẽ. Một số cái tên điển hình sử dụng nguyên lý này là SWT[1], MSER[2].

Trong điều kiện tương đối lý tưởng, dữ liệu sạch sẽ, ít có nhiều thì phương pháp này tỏ ra khá hiệu quả, độ chính xác cao và dễ thực hiện. Tuy nhiên, trong thực tế, rất khó để đảm bảo những điều kiện như vậy.

-Cách 2: Sử dụng kỹ thuật Deep Learning. Sử dụng kỹ thuật Deep Learning đã mang lại những tiến bộ vượt bậc trong bài toán phát hiện chữ trong ảnh. Các mô hình Deep Learning có khả năng học tập từ lượng dữ liệu lớn và biểu diễn các đặc trưng phức tạp của chữ một cách tự động, thay vì phải dựa vào các đặc trưng thủ công như trong các phương pháp truyền thống.

Các mô hình Deep Learning thường sử dụng các kiến trúc mạng nơ-ron sâu, chẳng hạn như CNN[3], để trích xuất các đặc trưng hình ảnh ở nhiều cấp độ khác nhau. Những đặc trưng này sau đó được sử dụng để phát hiện các vùng chứa chữ trong ảnh. Một số phương pháp nổi bật trong lĩnh vực này bao gồm:

-CTPN[4]: Đây là một trong những mô hình Deep Learning đầu tiên áp dụng cho phát hiện chữ, sử dụng cơ chế hồi quy để dự đoán các bounding box dọc theo chiều dài của từ hoặc cụm từ.

-EAST[5]: Mô hình này sử dụng một mạng CNN với các lớp phân đoạn để xác định vị trí của các từ trong ảnh. EAST có thể phát hiện cả văn bản ngang và nghiêng một cách hiệu quả và chính xác.

-CRAFT[6]: Phương pháp này tập trung vào việc phát hiện các vùng ký tự nhỏ lẻ và sau đó ghép chúng lại thành các từ hoàn chỉnh, giúp mô hình phát hiện chữ có hình dạng và kích thước đa dạng.

-DB[7]: Đây là một phương pháp phân đoạn văn bản mạnh mẽ, sử dụng kỹ thuật binarization có thể phân biệt để xác định chính xác biên giới của các vùng chứa chữ. Mô hình này cho phép phát hiện chính xác các từ hoặc ký tự trong ảnh, kể cả khi chúng có hình dạng phức tạp.

-FOTS[8]: Đây là một mô hình end-to-end kết hợp cả phát hiện và nhận dạng chữ trong một kiến trúc mạng duy nhất, giúp tối ưu hóa toàn bộ quá trình xử lý văn bản từ hình ảnh một cách nhanh chóng và hiệu quả.

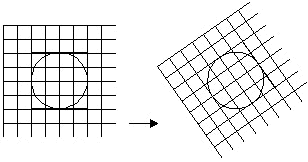
Kỹ thuật Deep Learning không chỉ giúp cải thiện độ chính xác của bài toán phát hiện chữ mà còn mở rộng khả năng của các hệ thống OCR hiện đại. Việc sử dụng các mô hình này đã trở thành xu hướng phổ biến, đặc biệt là khi đối mặt với những thách thức liên quan đến đa dạng hình dạng và kích thước của chữ trong các ngữ cảnh thực tế khác nhau.

## **2.3. Detection Boxes Rectify**

Trong quy trình của OCR, module chỉnh sửa hộp phát hiện văn bản( Detection Boxes Rectify) đóng vai trò quan trong và nằm giữa hai module chính text detection (phát hiện văn bản) và text recognition (nhận dạng văn bản). Chức năng của module này là chỉnh sửa các hộp chứa văn bản đã được phát hiện, đặc biệt trong trường hợp các hộp này có góc nghiêng không phải là 0 độ.

Sau khi văn bản được phát hiện bởi module text detection, kết quả là các hộp bao quanh các dòng văn bản trong ảnh. Tuy nhiên, do ảnh chụp có thể không được căn chỉnh chính xác, các dòng văn bản có thể bị nghiêng hoặc xoay ở các góc độ khác nhau. Trước khi đưa các hộp văn bản này vào module text recognition để nhận dạng nội dung, cần phải thực hiện thao tác chỉnh sửa để đảm bảo các dòng văn bản được đưa về trạng thái ngang chuẩn, thường là 0 độ.

Module Detection Boxes Rectify sử dụng phép biến đổi affine để thực hiện việc này. Cụ thể, module này chỉ hỗ trợ phân loại và chỉnh sửa góc văn bản trong khoảng lệnh 0 và 180 độ. Điều này có nghĩa là văn bản có thể được đưa về đúng hướng nếu nó bị xoay (180 độ<=), hoặc giữ nguyên nếu đã ở góc 0 độ. Sau khi các hộp văn bản được điều chỉnh, chúng sẽ được chuyển tiếp đến module text recognition để nhận dạng nội dung văn bản.

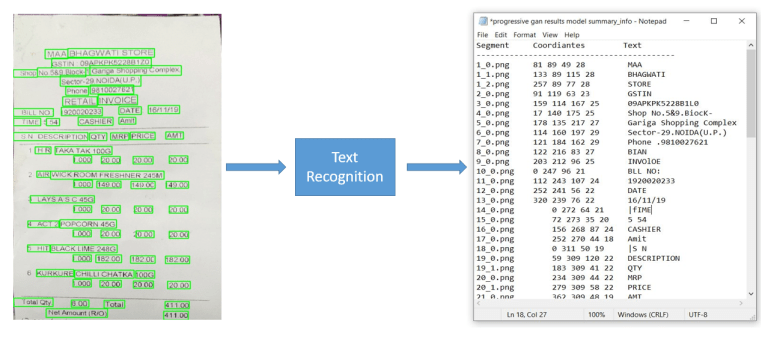


Hình 11.Biến đổi affine để thực hiện căn chỉnh góc hình ảnh

Như vậy, Detection Boxes Rectify là bước trung gian cần thiết để đảm bảo rằng các dòng văn bản được căn chỉnh chính xác trước khi tiến hành nhận dạng, giúp tăng độ chính xác của toàn bộ quy trình OCR.

## **2.4.Text Recognition**

Các ký tự/văn bản trong từng khu vực phát hiện ở bước bên trên sẽ được nhận diện cụ thể ở bước này.



Hình 12.Kết quả module Text Recognition

Tương tự như Text Detection, ở đây cũng có 2 phương pháp giải quyết là dùng kỹ thuật xử lý ảnh cơ bản và dùng kỹ thuật Deep Learning. Mỗi phương pháp đều có những ưu nhược điểm riêng, và sự lựa chọn giữa chúng phụ thuộc vào yêu cầu cụ thể của bài toán và môi trường ứng dụng.

Phương pháp đầu tiên sử dụng các kỹ thuật xử lý ảnh truyền thống để tách riêng từng ký tự ra khỏi nền trước khi đưa chúng qua một bộ phân lớp (classifier) để nhận diện. Quy trình này thường bao gồm các bước như ngưỡng hóa ảnh (thresholding), tách nền (background removal), và phân đoạn ký tự (character segmentation). Sau khi các ký tự được tách rời, chúng sẽ được đưa vào các thuật toán phân lớp như KNN, SVM, hoặc các mạng nơ-ron đơn giản để nhận dạng.

Tuy nhiên, cách tiếp cận này có một số hạn chế đáng kể. Do việc xử lý và nhận dạng được thực hiện ở mức độ ký tự đơn lẻ, độ chính xác của phương pháp này phụ thuộc rất nhiều vào chất lượng của bước Text Detection ban đầu và môi trường ảnh hưởng như nhiễu, biến dạng, hay các yếu tố khác trong ảnh. Kết quả là, trong những tình huống phức tạp hoặc khi chất lượng ảnh không cao, độ chính xác của phương pháp này có thể bị giảm sút đáng kể.

Khác với phương pháp truyền thống, Deep Learning đã chứng minh được tính ưu việt của nó trong nhiều ứng dụng khác nhau, bao gồm cả Text Recognition. Hai mô hình nổi bật được sử dụng trong Deep Learning cho nhiệm vụ nhận dạng văn bản là CRNN[9] và các mô hình dựa trên Attention.

CRNN kết hợp với CTC:

-CRNN là sự kết hợp giữa mạng nơ-ron tích chập (CNN) và mạng nơ-ron hồi quy (RNN). CNN được sử dụng để trích xuất các đặc trưng từ ảnh văn bản, sau đó RNN được áp dụng để xử lý chuỗi đặc trưng này, giúp nhận dạng các ký tự liên tiếp trong chuỗi mà không cần phải cắt rời từng ký tự riêng lẻ.

-CTC[10] là một phương pháp tối ưu hóa đầu ra của RNN, cho phép nhận dạng các chuỗi ký tự có chiều dài biến đổi mà không cần phải có nhãn cho từng ký tự trong chuỗi. Điều này rất hữu ích khi làm việc với các văn bản không phân cách rõ ràng giữa các ký tự.

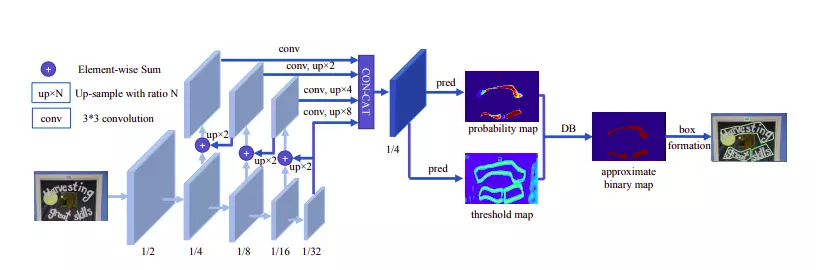
Attention-based Models:

-Mô hình dựa trên Attention đã mang lại một cuộc cách mạng trong lĩnh vực nhận dạng chuỗi ký tự. Thay vì xử lý toàn bộ chuỗi một cách tuần tự, mô hình Attention tập trung vào các phần cụ thể của chuỗi dữ liệu đầu vào khi thực hiện việc dự đoán, từ đó cải thiện độ chính xác và hiệu suất nhận dạng. Cơ chế Attention cho phép mô hình "chú ý" đến các vùng quan trọng của ảnh, giúp phân biệt và nhận dạng các ký tự trong những tình huống phức tạp hoặc bị nhiễu.

Phương pháp Deep Learning, với sự hỗ trợ của các mô hình như CRNN và Attention, đã vượt trội hơn hẳn so với các kỹ thuật xử lý ảnh truyền thống, đặc biệt là trong các ứng dụng yêu cầu độ chính xác cao và khả năng xử lý trong các môi trường phức tạp. Trong đó, CRNN là một trong những mô hình tiên tiến và phổ biến nhất, nhờ khả năng kết hợp giữa việc trích xuất đặc trưng mạnh mẽ từ ảnh và phân tích chuỗi ký tự liên tiếp một cách hiệu quả. CRNN đã chứng minh được sự ưu việt của mình trong việc xử lý các văn bản có cấu trúc phức tạp và là một giải pháp mạnh mẽ cho các bài toán nhận dạng văn bản trong OCR. Để hiểu rõ hơn về cách hoạt động và lý do tại sao CRNN lại đóng vai trò quan trọng trong hệ thống OCR hiện đại, tài liệu sẽ đi sâu vào phân tích chi tiết về mô hình này trong chương 4.

**CHƯƠNG 3: PHÁT HIỆN VĂN BẢN VỚI MÔ HÌNH DB**

Các mô hình trong bài toán phát hiện văn bản được nghiên cứu trước đây có quá trình xử lý rất phức tạp để lấy được các thông tin vị trí của đối tượng trong ảnh. Các phương pháp này sẽ đặt một ngưỡng cố định để phân biệt đâu là điểm ảnh nền, đâu là điểm ảnh đối tượng tử bản đồ xác suất được mô hình sinh ra. Việc đặt một ngưỡng cố định làm cho mô hình kém linh động và mất nhiều thời gian thử nghiệm để tìm ngưỡng phù hợp. Mô hình DB đã đề xuất một ý tưởng cho mô hình học luôn cách đặt giá giá trị ngưỡng này thông qua việc huấn luyện. Kiến trúc mô hình được mô tả trên hình dưới đây:



Hình 13.Kiến trúc mô hình DB

Mô hình sẽ nhận ảnh dầu vào và cho qua một bộ trích xuất đặc trưng sinh ra bản đồ đặc trưng F. Trong quá trình huấn luyện, bản đồ đặc trưng này sau đó được dùng để dự đoán bản đồ xác suất P và bản đồ ngưỡng T. Cuối cùng, sử dụng hai bản đồ này để tính ra bản đồ nhị phân được dùng cho quá trình hậu xử lý để lấy được dự đoán vị trí trong ảnh sau này. Còn trong quá trình kiểm tra, có thể lấy trực tiếp thông tin vị trí bản đồ xác suất hoặc bản đồ nhị phân.

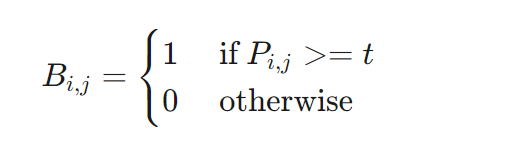
Kiến trúc DB có hai ưu điểm nổi bật so với các kiến trúc khác dựa trên mô hình phân đoạn:

-Kiến trúc DB có thể đạt độ chính xác cao với những bộ trích xuất đặc trưng nhẹ như ResNet 18 như trong bài báo.

-Có thể học được cách đặt ngưỡng động trên phương pháp gọi là Differentiable binarization.

## **3.1. Standard binarization**

Standard binarization hay còn gọi là phương pháp nhị phân hóa chuẩn được rất nhiều mô hình phát hiện văn bản sử dụng. Phương pháp này nhận đầu vào là một bản đồ xác suất P ∈ được sinh ra bởi bộ trích xuất đặc trưng và sinh ra bản đồ nhị phân có kích thước bằng với bản đồ xác suất (H, W lần lượt chỉ chiều cao và chiều rộng của bản đồ). Thuật toán của phương pháp này nhận vào các giá trị điểm ảnh của bản đồ xác suất. Các giá trị điểm ảnh nào lớn hơn một ngưỡng cố định thì giá trị điểm ảnh trên bản đồ nhị phân tương ứng nhận giá trị 1 thể hiện điểm ảnh đó thuộc về đối tượng chữ, còn 0 thì điểm ảnh đó không phải đối tượng chữ.

****

Ở đây t là một ngưỡng cố định, (i,j) là tọa độ vị trí điểm ảnh trên bản đồ.

Phương pháp này có nhược điểm là giá trị ngưỡng bị đặt cố định cho tất cả vị trí trên bản đồ. Do đó về mặc chủ quan sẽ gây nhầm lẫn ở các điểm ảnh mà giá trị ngưỡng cần thấp hoặc cao hơn và mất thời gian để tìm ra một ngưỡng phù hợp cho bài toán.

## **3.2. Differentiable binarization**

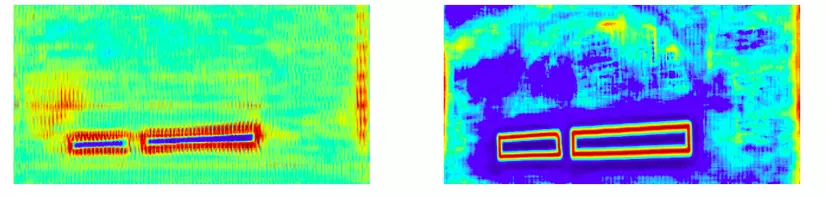
Để giải quyết vấn đề của phương pháp Standard binarization, bài báo đề xuất phương pháp Differentiable binarization. Do phương pháp mới này có thể thực hiện tính toán giá trị đạo hàm nên có thể tích hợp trực tiếp vào mô hình học sâu.

trong đó là bản đồ nhị phân, T bản đồ ngưỡng động được học từ mô hình, k là một chỉ số khuếch đại. Phương pháp này kết hợp sử dụng ngưỡng động được trình bày trong phần tiếp theo không chỉ giúp phát hiện đối tượng so với nền mà còn giúp tách các vùng chữ bị ghép liền vào với nhau.

## **3.3. Ngưỡng động (Adaptive threshold)**

Trong mô hình DB, một khía cạnh quan trọng là quá trình tạo ra bản đồ ngưỡng động (threshold map) từ các bản đồ đặc trưng (feature map) của mô hình phân đoạn. Bản đồ ngưỡng động này đóng vai trò thiết yếu trong việc xác định ranh giới của các đối tượng trong ảnh, đặc biệt là khi làm việc với các nhiệm vụ như phát hiện văn bản hoặc các đối tượng nhỏ và mảnh.

Cụ thể, trong quá trình phân đoạn, các bản đồ đặc trưng được sinh ra từ các tầng sâu của mạng nơ-ron. Thông thường, từ các bản đồ đặc trưng này, một bản đồ xác suất (probability map) cũng được tạo ra, thể hiện khả năng mỗi điểm ảnh (pixel) thuộc về một đối tượng cụ thể. Tuy nhiên, bản đồ ngưỡng động không chỉ dừng lại ở việc thể hiện xác suất này mà còn đi xa hơn trong việc xác định các điểm ảnh thuộc về cạnh ranh giới của đối tượng. Điều này rất quan trọng trong việc chính xác hóa ranh giới giữa các vùng khác nhau, giúp mô hình có thể nhận diện và phân đoạn đối tượng một cách tinh tế và chính xác hơn.



Hình 14.Bản đồ ngưỡng động

Bản đồ ngưỡng động xác định các điểm ảnh có khả năng nằm trên đường biên của đối tượng bằng cách thiết lập một ngưỡng động, cho phép linh hoạt trong việc xác định ranh giới, đặc biệt là với những đối tượng có hình dạng phức tạp hoặc nằm gần nhau. Quá trình này giúp tạo ra một sự phân đoạn sắc nét hơn, khi mô hình không chỉ nhận biết vùng mà còn nắm bắt rõ ràng các cạnh bao quanh đối tượng.

## **3.4. Hàm mất mát**

Hàm mất mát tác giả sử dụng là tổng có trọng số của ba hàm mất mát cho bản đồ xác suất , bản độ nhị phân ​ và bản đồ ngưỡng :

trong đó α,β lần lượt là trọng số ưu tiên của hàm mất mát cho bản đồ xác suất và bản đồ ngưỡng.

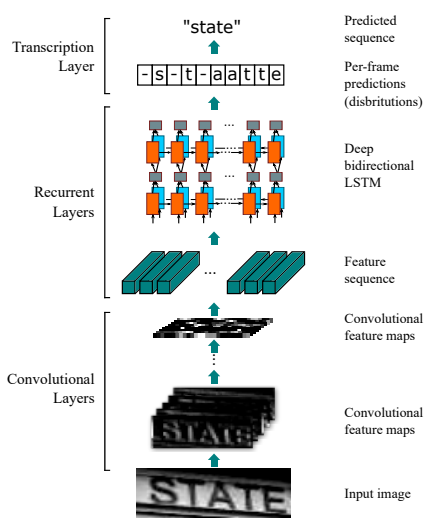
Do số lượng các điểm ảnh không phải chữ thường chiếm rất nhiều trong một ảnh nên để giảm hiện tượng mất cân bằng giữa các lớp, bài báo đề xuất sử dụng hàm mất mát BCE đồng thời hạn chế số lượng đối tượng không phải chữ được đưa vào hàm mất mát. Tập hợp các đối tượng không phải chữ được cho vào mất mát được gọi là ​ :

Hàm mất mát cho bản đồ ngưỡng được tính bằng tổng khoảng cách L1 giữa nhãn và dự đoán.

trong đó ​ là tập hợp các vị trí bên trong vùng chữ, ​ là nhãn cho bản đồ ngưỡng.

# **CHƯƠNG 4: NHẬN DIỆN VĂN BẢN VỚI MÔ HÌNH CRNN**

CRNN là một kiến trúc tiên tiến được thiết kế chuyên biệt để giải quyết nhiệm vụ Text Recognition trong bài toán OCR. Xuất hiện lần đầu trong bài báo "An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition" xuất bản năm 2015, CRNN đã nhanh chóng chứng minh được hiệu quả vượt trội của nó và cho đến nay vẫn được coi là một trong những mô hình hàng đầu cho việc nhận dạng văn bản từ hình ảnh.

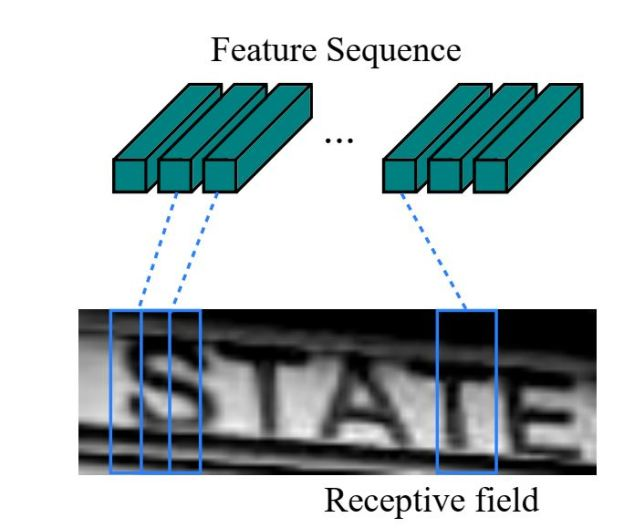


Hình 15.Kiến trúc mô hình CRNN

CRNN, như tên gọi, là sự kết hợp tinh tế giữa CNN và RNN, tạo nên một mô hình hoàn hảo cho việc nhận dạng văn bản từ hình ảnh. Việc sử dụng CNN là điều không thể thiếu khi xử lý hình ảnh, trong khi RNN lại đóng vai trò chủ chốt trong việc xử lý chuỗi văn bản. Kiến trúc của CRNN được chia thành ba phần chính, mỗi phần đóng vai trò quan trọng trong việc biến đổi một bức ảnh đầu vào thành văn bản tương ứng.

## **4.1 Convolutional Layers**

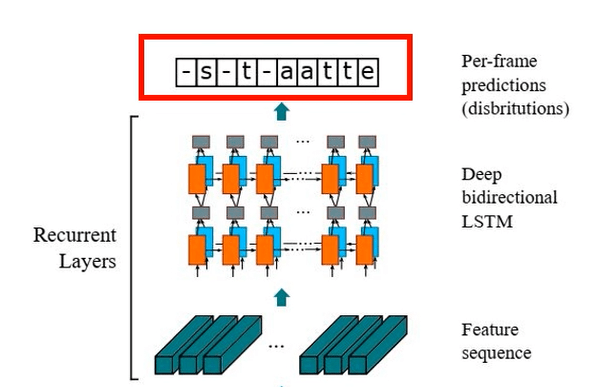
Ảnh đầu vào được đưa vào mô hình, nó sẽ đi qua một loạt các lớp tích chập, nơi các phép tính toán được thực hiện để trích xuất các đặc trưng quan trọng từ ảnh. Quá trình này tạo ra một loạt các bản đồ đặc trưng (Feature Maps), mỗi bản đồ biểu thị một khía cạnh cụ thể của ảnh gốc. Tuy nhiên, thay vì dừng lại ở việc chỉ tạo ra các bản đồ đặc trưng, CRNN tiến thêm một bước quan trọng: các bản đồ này sau đó được chia nhỏ thành một chuỗi các vector đặc trưng, được gọi là Feature Vectors. Mỗi vector đặc trưng này đại diện cho một đoạn nhỏ của bức ảnh ban đầu, tương ứng với một bước thời gian trong chuỗi, được gọi là Feature Sequence. Chính Feature Sequence này sẽ được đưa vào các lớp tiếp theo trong kiến trúc CRNN để xử lý và nhận dạng văn bản. Nhờ quá trình này, CRNN không chỉ trích xuất thông tin không gian từ ảnh mà còn chuyển đổi thông tin đó thành dạng chuỗi, chuẩn bị cho quá trình xử lý tiếp theo bởi các lớp RNN.



Hình 16.Kết quả Convolutional Layers của ảnh đầu vào

## **4.2 Recurrent Layers**

Sau khi Feature Sequence được tạo ra từ các lớp tích chập, nó sẽ được đưa vào các lớp LSTM hai chiều (Bidirectional LSTM). Các lớp này có khả năng ghi nhớ thông tin từ cả hai hướng thời gian, cho phép mô hình nắm bắt các ngữ cảnh xung quanh mỗi bước thời gian trong chuỗi.



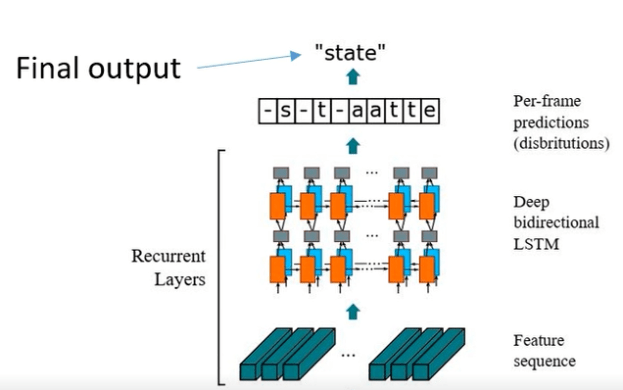
Hình 17.Feature sequence được đưa qua Recurrent Layers

Khi Feature Sequence đi qua các lớp LSTM, nó sẽ sinh ra một chuỗi các ký tự, trong đó mỗi ký tự tương ứng với một bước thời gian trong Feature Sequence. Về lý thuyết, đây chính là chuỗi văn bản mà mô hình cần xác định từ bức ảnh đầu vào. Tuy nhiên, trong thực tế, việc chia nhỏ các bản đồ đặc trưng thành các vector đặc trưng không phải lúc nào cũng diễn ra một cách hoàn hảo. Điều này dẫn đến việc mỗi vector đặc trưng không nhất thiết phải chứa đúng một ký tự duy nhất cần nhận diện.

Do đó, chuỗi đầu ra từ các lớp LSTM thường gặp phải những vấn đề như lặp lại ký tự, thiếu ký tự, hoặc thậm chí là các ký tự không cần thiết. Đây là lý do tại sao chuỗi kết quả từ LSTM, mặc dù chứa thông tin về văn bản trong ảnh, nhưng vẫn cần được xử lý thêm để đảm bảo tính chính xác và độ tin cậy của văn bản được nhận diện. Quá trình này sẽ được thực hiện trong phần cuối cùng của kiến trúc CRNN.

## **4.3 Transcription Layers**

Nhiệm vụ của phần này là xử lý đầu ra của Recurrent Layers, sắp xếp lại các ký tự, loại bỏ các lỗi tồn tại (alignment) để đưa ra kết quả cuối cùng.

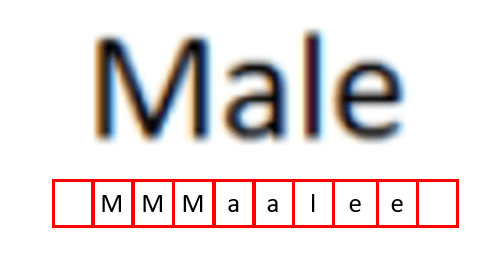


Hình 18.Kết quả Transcription Layers

Có một số phương pháp tiếp cận được áp dụng trong Transcription Layers để thực hiện quá trình này:

Cách 1 - Quy định mỗi ký tự tương ứng với một số lượng cố định các TimeStep: Phương pháp này rất đơn giản và dễ thực hiện, trong đó mỗi ký tự trong văn bản đầu ra sẽ được ánh xạ vào một số lượng cố định các TimeStep. Ví dụ, nếu có 10 TimeStep và đầu ra dự kiến là từ "STATE", ta có thể sắp xếp các TimeStep như sau: “S, S, T, T, A, A, T, T, E, E”. Tuy nhiên, cách này chỉ hoạt động tốt khi kích thước, kiểu chữ, và các yếu tố khác của văn bản đầu vào không thay đổi. Do đó, phương pháp này thường không linh hoạt và dễ gặp phải vấn đề khi xử lý các văn bản đa dạng.

Cách 2 - Đánh nhãn cho mỗi TimeStep: Trong phương pháp này, mỗi TimeStep trong Feature Sequence sẽ được gán một nhãn cụ thể, tương ứng với ký tự cần nhận diện. Mô hình sau đó sẽ được huấn luyện với dữ liệu đã được gán nhãn này, giúp nó học cách liên kết mỗi TimeStep với ký tự đúng. Phương pháp này thường mang lại kết quả chính xác hơn so với cách 1, nhưng nhược điểm lớn là việc tạo dữ liệu gán nhãn đòi hỏi rất nhiều thời gian và công sức.

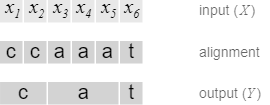


Hình 19.Đánh nhãn cho mỗi Timestamp

Cách 3 - Sử dụng Connectionist Temporal Classification (CTC): CTC là một kỹ thuật được phát triển từ năm 2006, đặc biệt để giải quyết vấn đề sắp xếp (Alignment) giữa đầu ra của RNN và kết quả văn bản cuối cùng. CTC được sử dụng rộng rãi không chỉ trong OCR mà còn trong các lĩnh vực khác như nhận dạng giọng nói. Phương pháp này giúp tự động xử lý các sự khác biệt về độ dài giữa đầu ra của RNN và chuỗi ký tự thực tế mà không cần phải đánh nhãn thủ công từng TimeStep như cách 2. CTC hoạt động bằng cách cho phép mô hình dự đoán các chuỗi có thể khác nhau về độ dài so với chuỗi gốc và tự động sắp xếp chúng lại để tạo ra văn bản cuối cùng chính xác nhất

## **4.4. Connectionist Temporal Classification**

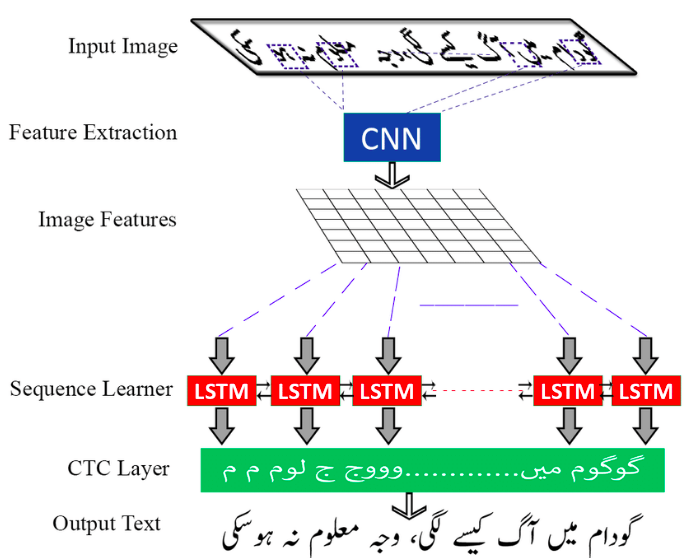
CTC thực chất là một hàm mất mát (Loss Function) đặc biệt được sử dụng trong quá trình huấn luyện các mô hình học sâu (Deep Learning). Điểm mạnh nổi bật của CTC là khả năng tìm ra cách ánh xạ (alignment) hiệu quả giữa một đầu vào X (thường là một chuỗi các đặc trưng hoặc Feature Sequence) và đầu ra Y (chuỗi ký tự hoặc văn bản cần nhận diện).



Hình 20.Kết quả của X sau khi đưa qua CTC

Khác với các phương pháp truyền thống, CTC không yêu cầu dữ liệu phải được gán nhãn cụ thể cho từng bước thời gian (TimeStep). Điều này có nghĩa là không cần cung cấp cho mô hình thông tin chi tiết về ký tự nào tương ứng với mỗi TimeStep trong chuỗi đầu vào. Thay vào đó, CTC hoạt động dựa trên việc tính toán xác suất cho mọi khả năng ánh xạ từ chuỗi đầu vào X sang chuỗi đầu ra Y, cho phép mô hình tự động học cách sắp xếp các ký tự một cách chính xác mà không cần sự can thiệp chi tiết từ người dùng.

Điều duy nhất mà CTC yêu cầu là một đầu vào bao gồm hình ảnh (đã được chuyển đổi thành ma trận đặc trưng thông qua các lớp CNN) và văn bản tương ứng với hình ảnh đó. Với thông tin này, CTC sẽ tối ưu hóa quá trình học bằng cách đưa ra một sự kết hợp tốt nhất giữa chuỗi đặc trưng và chuỗi ký tự, giúp mô hình học cách nhận diện văn bản từ hình ảnh một cách chính xác và hiệu quả.



Trong đó là kết quả đầu ra cuối cùng mà ta mong đợi, nó càng giống với nhãn (GT) càng tốt. Nhờ khả năng linh hoạt này, CTC trở thành một công cụ vô cùng mạnh mẽ trong các bài toán liên quan đến nhận dạng ký tự quang học (OCR) và nhận dạng giọng nói, nơi mà sự khác biệt giữa độ dài của chuỗi đầu vào và đầu ra thường là một thách thức lớn.

# **CHƯƠNG 5: THỬ NGHIỆM VÀ ĐÁNH GIÁ**

Đối với mô hình phát hiện văn bản, tập dữ liệu gồm 97.000 hình ảnh được sử dụng để huấn luyện và 500 hình ảnh được sử dụng để kiểm tra. Các tập dữ liệu công khai bao gồm LSVT, RCTW-17, MTWI 2018, CASIA-10K, SROIE, MLT 2019, BDI, MSRA-TD500 và CCPD 2019. Phần lớn các hình ảnh này là hình ảnh văn bản tài liệu.

|  |  |  |
| --- | --- | --- |
| **Mô hình** | **Số lượng dữ liệu huấn luyện** | **Số lượng dữ liệu kiểm ta** |
| Phát hiện văn bản(DB) | 97000 | 500 |
| Nhận diện văn bản(CRNN) | 17900000 | 18700 |

Bảng 1.Chi tiết về tập dữ liệu đào tạo và kiểm tra

Trong quá trình huấn luyện mô hình, kĩ thuật tối ưu Adam được sử dụng để huấn luyện tất cả các mô hình và áp dụng phương pháp giảm learning\_rate dựa trên hàm Cosine. Learning\_rate ban đầu, batch\_size, số epoch cho từng mô hình được trình bày trong bảng 2.

|  |  |  |  |
| --- | --- | --- | --- |
| **Mô hình** | **Learning\_rate ban đầu** | **Batch\_size** | **Số lượng Epoch** |
| Phát hiện văn bản(DB) | 0.001 | 16 | 60 |
| Nhận diện văn bản(CRNN) | 0.001 | 1024 | 100 |

Bảng 2.Chi tiết về các tham số huấn luyện

Sau khi các mô hình được huấn luyện, phương pháp FPGM Pruner và PACT để giảm kích thước mô hình. Trong gian đoạn thử nghiệm, HMean được sử dụng để đánh giá hiệu suất của mô hình DB, Accuracy để đánh giá độ chính xác của mô hình CRNN và F-score được sử dụng để đánh giá hiệu suất của toàn hệ thống OCR.

Bảng 3 so sánh hiệu suất của các kiến trúc khác nhau trong mô đun phát hiện văn bản. Trong các kiến trúc được thử nghiệm, MobileNetV3 largex0.5 thể hiện kết cả tốt khi cân bằng giữa độ chính xác và hiệu suất.

|  |  |  |  |
| --- | --- | --- | --- |
| **Backbone** | **HMean** | **Model Size(M)** | **Inference Time(CPU,ms)** |
| MobileNetV3\_large\_x1 | 0.6463 | 16 | 447 |
| MobileNetV3\_large\_x0.5 | 0.6127 | 7 | 406 |
| MobileNetV3\_large\_x0.35 | 0.5935 | 5.4 | 367 |
| MobileNetV3\_small\_x1 | 0.5919 | 7.5 | 380 |

Bảng 3.So sánh hiệu suất của 5 kiến trúc khác nhau trong mô đun nhận diện văn bản

Bảng 4 thể hiện hiệu quả của FPGM pruner khi sử dụng trong mô đun phát hiện văn bản. Khi sử dụng FPGM pruner, kích thước mô hình giảm 46.2% và thời gian thực thi giảm 18.9% mặc dù chỉ số HMean giảm nhẹ. Điều này cho thấy FPGM pruner có khả năng tăng tính hiệu quả của mô hình phát hiện văn bản.

|  |  |  |
| --- | --- | --- |
| **HMean** | **Model Size(M)** | **Inference Time(CPU,ms)** |
| 0.6239 | 2.6 | 164 |
| 0.6169 | 1.4 | 133 |

Bảng 4.Hiệu suất của mô hình phát hiện văn bản khi sử dụng FPGM

Đối với mô hình nhận diện văn bản, tập dữ liệu bao gồm 17,9 triệu hình ảnh được sử dụng để huấn luyện và 18700 hình ảnh được sử dụng để kiểm tra. Các tập dữ liệu công khai bao gồm LSVT, RCTW-17, MTWI 2018, và CCPD 2019. Bảng 5 so sánh hiệu suất của các kiến trúc khác nhau dùng trong mô đun nhận diện văn bản. Độ chính xác, kích thước mô hình và thời gian thực thi của các trường hợp sử dụng các kiến trúc khác nhau của mô hình MobileNetV3 thay đổi đáng kể. MobileNetV3\_small\_x0.5 được chọn để cân bằng giữa độ chính xác và hiệu suất của mô hình.

|  |  |  |  |
| --- | --- | --- | --- |
| **Backbone** | **Accuracy** | **Model Size(M)** | **Inference Time(CPU, ms)** |
| MobileNetV3\_large\_x0.35 | 0.6288 | 22 | 17 |
| MobileNetV3\_large\_x0.5 | 0.6556 | 23 | 17.27 |
| MobileNetV3\_small\_x1 | 0.6933 | 28 | 19.15 |

Bảng 5.So sánh hiệu suất của 5 kiến trúc khác nhau trong mô đun nhận diện văn bản

Bảng 6 biểu diễn kết quả khi sử dụng PACT cho nhận diện văn bản. Khi áp dụng PACT, kích thước mô hình giảm 67,39% và thời gian thực thi giảm 8,3%. Tuy thời gian thực thi không cải thiện rõ rệt, tuy nhiên độ chính xác được cải thiện đáng kể. Vì vậy, PACT là một chiến lược hiệu quả để giảm kích thước mô hình của hệ thống nhận dạng văn bản.

|  |  |  |
| --- | --- | --- |
| **Accuracy** | **Model Size(M)** | **Inference Time(SD 855,ms)** |
| 0.6518 | 4.6 | 12 |
| 0.674 | 1.5 | 11 |

Bảng 6.Hiệu suất của mô hình nhận diện văn bản khi sử dụng PACT

# **KẾT LUẬN**

Mô hình DB đã cung cấp giải pháp hiệu quả cho việc phân tách văn bản khỏi nền, giúp cải thiện khả năng phát hiện văn bản trong các điều kiện phức tạp. Đồng thời, mô hình CRNN kết hợp các đặc trưng của mạng nơ-ron tích chập và mạng nơ-ron hồi tiếp, cho phép nhận diện và phân tích chuỗi văn bản với độ chính xác cao hơn.

Đồng thời, việc áp dụng các phương pháp tối ưu hóa FPGM Pruner và PACT đã cho thấy những cải thiện đáng kể trong việc giảm kích thước mô hình và nâng cao hiệu suất. FPGM Pruner giúp loại bỏ các bộ lọc không cần thiết, giảm thiểu khối lượng tính toán mà không làm giảm chất lượng nhận diện văn bản. Đồng thời, PACT Quantization cho phép nén mô hình hiệu quả thông qua việc áp dụng các quy tắc xác định ngưỡng và lượng tử hóa, giúp cải thiện tốc độ thực thi mà vẫn duy trì độ chính xác cao.

Việc kết hợp các phương pháp tối ưu hóa này không chỉ giúp giảm yêu cầu về tài nguyên tính toán mà còn làm cho mô hình OCR trở nên phù hợp hơn với các ứng dụng di động và hệ thống có tài nguyên hạn chế. Mặc dù các công nghệ này đã đạt được nhiều thành tựu, vẫn còn những thách thức trong việc áp dụng chúng vào các điều kiện thực tế, chẳng hạn như xử lý các kiểu phông chữ đa dạng và điều kiện ánh sáng không đồng đều. Tuy nhiên, những thách thức này cũng mở ra nhiều cơ hội để cải tiến và phát triển thêm các giải pháp OCR. Hướng tới tương lai, việc tiếp tục tối ưu hóa các mô hình và áp dụng các phương pháp giảm kích thước như FPGM Pruner và PACT Quantization sẽ đóng vai trò quan trọng trong việc đáp ứng nhu cầu ngày càng cao của các ứng dụng thực tế. Những nghiên cứu và phát triển tiếp theo có thể tập trung vào việc nâng cao khả năng xử lý đa ngôn ngữ và cải thiện tốc độ thực thi, đồng thời mở rộng khả năng áp dụng của công nghệ OCR trong các lĩnh vực khác nhau.

# **TÀI LIỆU THAM KHẢO**

[1]Boris Epshtein. (2010, 11). *Detecting Text in Natural Scenes with Stroke Width Transform*.

[2]Michael Donoser. (2006, 6). *Efﬁcient Maximally Stable Extremal Region (MSER) Tracking*.

[3]Purwono. (2022, 4). *Understanding of Convolutional Neural Network (CNN): A Review*.

[4]Zhi Tian. (2016, 10). *Detecting Text in Natural Image with Connectionist Text Proposal Network*.

[5]Xinyu Zhou. (2017, 7). *EAST: An Efﬁcient and Accurate Scene Text Detector*.

[6]Youngmin Baek. (2019, 6). *Character Region Awareness for Text Detection*.

[7]Minghui Liao. (2020, 4). *Real-Time Scene Text Detection with Differentiable Binarization*.

[8]Xuebo Liu. (2018, 6). *FOTS: Fast Oriented Text Spotting with a Unified Network*.

[9]Ruishuang Wang. (2019, 6). *Convolutional Recurrent Neural Networks for Text Classification*.

[10]Alex Graves. (2006, 1). *Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural 'networks*.