X

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

KHOA CÔNG NGHỆ THÔNG TIN 1

*****

XỬ LÝ ẢNH

HỆ THỐNG NHẬN DIỆN CHỮ VIẾT TAY TIẾNG VIỆT

Giảng viên: Đào Thị Thúy Quỳnh

Nhóm bài tập: 16

Nhóm sinh viên: Nguyễn Mạnh Hùng

Đoàn Minh Hiển

Nguyễn Thanh Tùng

Hà Nội, 2024

Table of Contents

PHẦN MỞ ĐẦU....3
1. Mục đích của bài toán....3
2. Phương pháp tiếp cận....4
CHƯƠNG 1: TỔNG QUAN MÔ HÌNH....6
1.1. Mô tả kiến trúc....6
1.1.1. Tóm tắt cấu trúc chính của mô hình....6
1.1.2. Các thành phần chính....7
1.1.3. Vai trò của từng khối trong hệ thống:....8
1.2. Ý tưởng chính....9
CHƯƠNG 2: DỮ LIỆU VÀ TIỀN XỬ LÝ DỮ LIỆU....11
2.1. Nguồn dữ liệu....11
2.2. Phân tích dữ liệu ban đầu....11
2.3. Chia tập dữ liệu....12
2.4. Tiền xử lý dữ liệu....12
2.4.1. Mục tiêu....12
2.4.2. Các bước xử lý hình ảnh....13
2.4.3. Xử lý nhãn....14
2.5. Kết quả của tiền xử lý....15
2.6. Ý nghĩa của tiền xử lý....15
CHƯƠNG 3: HUẤN LUYỆN VÀ TRIỂN KHAI....17
3.1. Quy trình huấn luyện....17
3.1.1. Thông số chính trong quá trình huấn luyện....17
3.1.2. Chi tiết thuật toán tối ưu (Adam)....17
3.2. Hàm mất mát....18
3.2.1. Giải thích CTC Loss (Connectionist Temporal Classification)....18
3.2.2. Lý do sử dụng CTC loss....19
3.2.3. Vai trò của input_length và label_length....19
3.3. Đánh giá mô hình....20
CHƯƠNG 4: KẾT QUẢ....21
4.1. Hiệu năng....21
4.2. Thảo luận....21
CHƯƠNG 5: XÂY DỰNG GIAO DIỆN....23
5.1. Mục tiêu xây dựng giao diện....23
5.2. Công nghệ sử dụng....24
5.3. Quy trình xử lý trong giao diện....24
5.4. Các thành phần giao diện chính....25
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN....27
TÀI LIỆU THAM KHẢO....28

PHẦN MỞ ĐẦU

1. Mục đích của bài toán

Nhận dạng ký tự quang học (Optical Character Recognition - OCR) là một nhánh quan trọng trong lĩnh vực xử lý ảnh số, được sử dụng để chuyển đổi các hình ảnh chứa văn bản viết tay hoặc in ấn thành dữ liệu ký tự số hóa. Với OCR chữ viết tay, độ phức tạp của bài toán tăng lên đáng kể do sự không đồng nhất về kiểu chữ, hình dạng ký tự, nét chữ của từng người viết và sự hiện diện của nhiễu từ chất lượng hình ảnh không đồng đều. Những yếu tố này đặt ra thách thức lớn đối với các thuật toán nhận dạng, đặc biệt khi dữ liệu đầu vào không theo quy tắc cố định.

Công nghệ OCR chữ viết tay đã và đang có nhiều ứng dụng thực tiễn trong cuộc sống. Một số ứng dụng nổi bật bao gồm:

- **Số hóa tài liệu viết tay:** Chẳng hạn, biểu mẫu, đơn từ, tài liệu lịch sử hoặc các ghi chú viết tay có thể được lưu trữ và tìm kiếm dễ dàng hơn trong dạng số hóa.
- **Tự động hóa quy trình xử lý văn bản:** Hỗ trợ các hệ thống quản lý thông tin trong việc nhập liệu tự động hoặc phân loại dữ liệu, giảm thiểu lỗi con người.
- **Hỗ trợ công nghệ giao tiếp thông minh:** Nhận diện chữ viết trên các bề mặt cảm ứng như máy tính bảng, điện thoại thông minh hoặc các thiết bị nhập liệu tương tự, từ đó nâng cao trải nghiệm người dùng.

Tiếng Việt mang những đặc thù riêng biệt, làm tăng độ phức tạp cho bài toán OCR. Đầu tiên, hệ thống chữ cái tiếng Việt bao gồm **29 chữ cái chính**, kèm theo các **dấu thanh** (sắc, huyền, hỏi, ngã, nặng) và dấu phụ như trong các chữ cái "ă", "â", "ê", "ô", "ơ", "ư". Điều này dẫn đến không gian ký tự lớn hơn so với các ngôn ngữ như tiếng Anh. Hơn nữa:

- **Chữ ghép**: Một số chữ như "ch", "nh", "th" yêu cầu mô hình nhận dạng chính xác trong bối cảnh ngữ nghĩa.
- **Khác biệt cá nhân trong chữ viết tay:** Nét chữ của mỗi người có thể khác nhau đáng kể, từ cách viết các nét cơ bản cho đến độ dày, góc nghiêng, hoặc thậm chí phong cách riêng.

- **Nhiễu trong dữ liệu:** Các yếu tố như mờ nhòe, chói sáng, hoặc chất lượng hình ảnh kém làm tăng độ khó của bài toán, yêu cầu mô hình có khả năng xử lý dữ liệu đa dạng và không đồng nhất.

Với những thách thức này, việc nghiên cứu và phát triển mô hình OCR chữ viết tay tiếng Việt không chỉ là một bài toán khoa học mà còn mở ra tiềm năng lớn trong ứng dụng thực tiễn, giúp giải quyết những vấn đề trong cuộc sống và công việc hàng ngày.

2. Phương pháp tiếp cận

Để giải quyết bài toán nhận dạng chữ viết tay tiếng Việt, chúng tôi sử dụng mạng học sâu, một công nghệ đã mang lại bước tiến vượt bậc trong lĩnh vực nhận dạng ký tự quang học (OCR). Deep Learning đặc biệt hiệu quả trong việc xử lý các dữ liệu không chuẩn, như chữ viết tay, nhờ khả năng tự động học và trích xuất các đặc trưng phức tạp từ dữ liệu đầu vào mà không cần sự can thiệp thủ công. Trong mô hình của chúng tôi, chúng tôi kết hợp các mạng nơ-ron tích chập (CNN) để trích xuất đặc trưng từ hình ảnh và các mạng nơ-ron hồi tiếp (RNN), đặc biệt là LSTM (Long Short-Term Memory), để xử lý chuỗi ký tự đầu ra. Mạng CNN giúp trích xuất các đặc trưng hình ảnh quan trọng, chẳng hạn như các nét chữ, hình dạng ký tự và dấu thanh, từ dữ liệu hình ảnh chữ viết tay. Khả năng của CNN trong việc nhận diện các đặc điểm không gian trong hình ảnh là rất quan trọng, giúp giảm thiểu sự phức tạp trong việc thiết kế các phương pháp trích xuất đặc trưng thủ công.

Bên cạnh đó, chúng tôi sử dụng LSTM hai chiều (Bidirectional LSTM), một biến thể mạnh mẽ của mạng RNN, để hiểu và xử lý chuỗi ký tự đầu ra. LSTM có khả năng ghi nhớ thông tin qua các bước thời gian, giúp mô hình giải quyết các vấn đề về sự biến mất gradient. Việc sử dụng LSTM hai chiều mang lại lợi thế lớn, vì mô hình có thể học từ cả hai hướng—từ trái qua phải và từ phải qua trái. Điều này đặc biệt hữu ích trong nhận dạng chữ viết tay tiếng Việt, nơi cấu trúc ngữ nghĩa có thể phụ thuộc vào các ký tự trước và sau. Hiểu ngữ cảnh từ cả hai hướng giúp mô hình nhận dạng chính xác hơn, đặc biệt là với các ký tự phức tạp và các dấu thanh.

Để giải quyết vấn đề về sự không khớp giữa độ dài chuỗi đầu vào (hình ảnh) và độ dài chuỗi đầu ra (ký tự), chúng tôi sử dụng hàm mất mát CTC (Connectionist Temporal Classification). CTC là một phương pháp phổ biến trong OCR, giúp mô hình nhận dạng ký tự

mà không yêu cầu một điểm căn chỉnh chính xác giữa các chuỗi đặc trưng và chuỗi ký tự. Điều này cho phép mô hình học và dự đoán chính xác các ký tự, dù chuỗi đầu ra có độ dài thay đổi so với chuỗi đầu vào. Tất cả các yếu tố này kết hợp lại giúp kiến trúc mạng của chúng tôi có thể xử lý linh hoạt và chính xác các đặc thù của chữ viết tay tiếng Việt, bao gồm dấu thanh, ký tự ghép và sự đa dạng trong cách viết của từng cá nhân, từ đó cải thiện khả năng nhận dạng và giảm thiểu lỗi trong quá trình nhận diện.

CHƯƠNG 1: TỔNG QUAN MÔ HÌNH

1.1. Mô tả kiến trúc

1.1.1. Tóm tắt cấu trúc chính của mô hình

Mô hình nhận dạng chữ viết tay (OCR) mà chúng tôi phát triển được thiết kế để nhận diện các ký tự trong hình ảnh chữ viết tay, chuyển đổi chúng thành văn bản có thể đọc được. Cấu trúc chính của mô hình bao gồm ba thành phần cốt lõi: Khối trích xuất đặc trưng (Feature Extraction Block), Khối mô hình hóa chuỗi (Sequence Modeling Block) và Hàm mất mát CTC (Connectionist Temporal Classification).

Khối trích xuất đặc trưng (Feature Extraction Block): Đây là phần đầu tiên trong mô hình, có nhiệm vụ trích xuất các đặc trưng từ hình ảnh đầu vào. Hình ảnh chữ viết tay được chuyển qua các lớp Convolutional Neural Network (CNN), cho phép mô hình tự động học và nhận diện các đặc trưng như nét chữ, dấu thanh, độ dày mỏng của chữ viết và các chi tiết cục bộ khác. Quá trình này giúp mô hình có thể nhận diện hình ảnh chữ viết tay dù có sự khác biệt lớn về kiểu chữ và cách viết. Bằng cách sử dụng các lớp CNN, mô hình không chỉ tự động học được những đặc trưng quan trọng mà còn giảm thiểu sự can thiệp thủ công trong việc xử lý dữ liệu.

Khối mô hình hóa chuỗi (Sequence Modeling Block): Sau khi các đặc trưng từ hình ảnh được trích xuất qua các lớp CNN, chúng sẽ được đưa vào một khối Long Short-Term Memory (LSTM) hai chiều. Khối này có vai trò quan trọng trong việc học mối quan hệ giữa các ký tự liên tiếp trong chuỗi văn bản. LSTM hai chiều giúp mô hình hiểu ngữ cảnh của từng ký tự không chỉ từ trái sang phải (theo thời gian), mà còn ngược lại, từ phải sang trái. Điều này rất quan trọng trong bài toán OCR, đặc biệt là đối với những ngữ cảnh phức tạp như nhận dạng dấu thanh và chữ ghép trong tiếng Việt. Việc mô hình hóa chuỗi ký tự theo cách này giúp mô hình nhận dạng chính xác hơn và tránh được các sai sót do ngữ cảnh của ký tự bị thiếu.

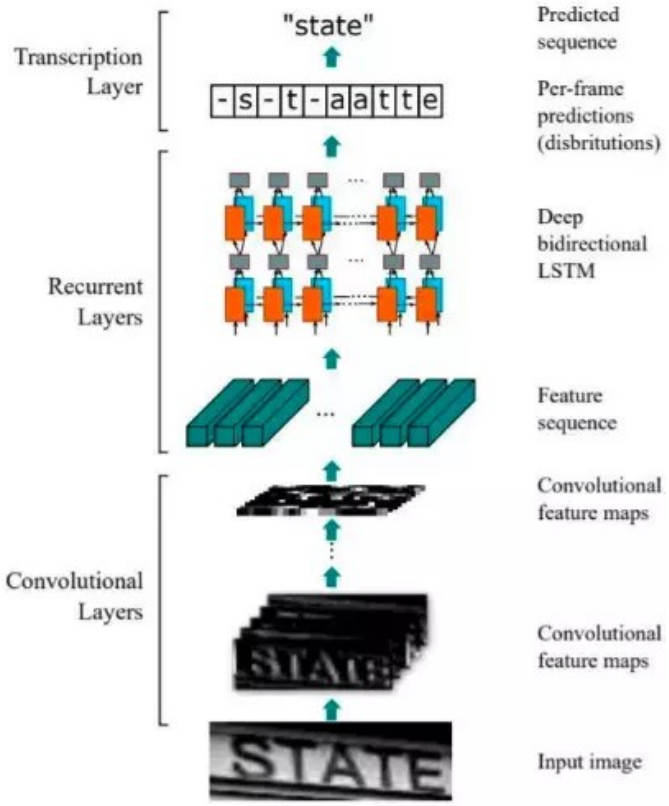
Hàm mất mát CTC (Connectionist Temporal Classification): Trong các mô hình OCR, một trong những thách thức lớn là sự không khớp giữa độ dài chuỗi đầu vào (hình ảnh) và chuỗi đầu ra (chuỗi ký tự). Đặc biệt là trong các trường hợp chữ viết tay, độ dài của chuỗi đặc trưng (từ CNN) có thể dài hơn hoặc ngắn hơn so với chuỗi văn bản thực tế mà mô hình cần nhận diện. Hàm mất mát CTC được sử dụng để giải quyết

vấn đề này. CTC cho phép mô hình ánh xạ đầu vào và đầu ra mà không cần phải căn chỉnh chính xác độ dài giữa chúng. Thay vào đó, CTC học cách tìm ra chuỗi ký tự đúng bằng cách tối ưu hóa khả năng dự đoán của mô hình trong khi cho phép có sự linh hoạt trong việc dự đoán các khoảng trống hoặc ký tự dư thừa. Điều này giúp mô hình trở nên linh hoạt hơn khi xử lý các chuỗi có độ dài thay đổi và đảm bảo chất lượng nhận dạng tốt hơn trong các tình huống phức tạp.

Tổng thể, cấu trúc của mô hình này được xây dựng để có thể tự động trích xuất và học các đặc trưng hình ảnh, hiểu mối quan hệ giữa các ký tự trong chuỗi, và cuối cùng là sử dụng CTC để tạo ra những dự đoán chính xác, linh hoạt từ chuỗi đặc trưng.

1.1.2. Các thành phần chính

Conv2D là lớp đầu tiên trong mô hình và có vai trò quan trọng trong việc trích xuất các đặc trưng cục bộ từ hình ảnh. Trong bài toán nhận dạng chữ viết tay, Conv2D giúp phát hiện các yếu tố cơ bản như nét chữ, dấu thanh, và các chi tiết cấu trúc của từng ký tự. Điều này rất quan trọng vì mỗi ký tự trong chữ viết tay có thể có hình dạng khác nhau, và lớp Conv2D tự động học các đặc trưng mà không cần sự can thiệp thủ công, giúp mô hình trở nên linh hoạt và mạnh mẽ trong việc nhận dạng các hình thức chữ viết tay đa dạng.



MaxPooling là một kỹ thuật được sử dụng để giảm kích thước dữ liệu mà vẫn giữ lại những đặc trưng quan trọng nhất. Qua đó, nó giúp tăng tốc quá trình tính toán, giảm độ phức tạp và giảm thiểu thông tin dư thừa trong các lớp trước đó. Bằng cách này, MaxPooling không chỉ giúp mô hình nhanh chóng xử lý các ảnh có độ phân giải cao mà còn giúp tránh hiện tượng quá khớp (overfitting), đồng thời tập trung vào những đặc trưng có ý nghĩa nhất đối với quá trình nhận dạng.

Batch Normalization được áp dụng để chuẩn hóa các giá trị đầu ra của các lớp chập. Việc chuẩn hóa này giúp ổn định quá trình huấn luyện và giảm thiểu sự thay đổi lớn giữa các lớp, từ đó giúp mô hình học nhanh hơn và hội tụ hiệu quả hơn. Ngoài ra, Batch Normalization còn giúp giảm thiểu vấn đề vanishing/exploding gradients, giúp cải thiện độ chính xác và tốc độ hội tụ trong quá trình huấn luyện, đặc biệt là với các mô hình phức tạp như OCR.

Bidirectional LSTM là một thành phần quan trọng trong khối mô hình hóa chuỗi của mô hình OCR. Khác với LSTM đơn chiều, LSTM hai chiều học từ cả hai hướng: từ trái sang phải và ngược lại. Điều này giúp mô hình hiểu rõ hơn về ngữ cảnh tổng thể của chuỗi ký tự, một yếu tố quan trọng trong nhận dạng chữ viết tay. Việc nắm bắt thông tin từ cả hai chiều giúp mô hình giảm thiểu các lỗi nhận dạng khi xử lý các ký tự có tính ngữ cảnh phức tạp, đặc biệt trong tiếng Việt với các ký tự ghép và dấu thanh.

Cuối cùng, CTC Loss (Connectionist Temporal Classification) đóng vai trò quan trọng trong việc giải quyết vấn đề về độ dài không đồng nhất giữa đầu vào (hình ảnh) và đầu ra (chuỗi ký tự). Trong OCR, đặc biệt là với chữ viết tay, các chuỗi đầu vào có thể có độ dài khác nhau, và CTC Loss giúp ánh xạ các đặc trưng từ hình ảnh đầu vào với chuỗi ký tự đầu ra mà không yêu cầu độ dài của chúng phải khớp. Điều này giúp mô hình xử lý các khoảng trống, ký tự thiếu hoặc thừa trong chuỗi dự đoán, cải thiện độ chính xác khi nhận dạng chữ viết tay.

1.1.3. Vai trò của từng khối trong hệ thống:

Khối chập (CNN Block) đóng vai trò quan trọng trong việc trích xuất các đặc trưng từ hình ảnh đầu vào. Khi xử lý hình ảnh chữ viết tay, các ký tự có hình dạng và cấu trúc phức tạp, với sự thay đổi lớn giữa các phong cách viết. Khối CNN giúp mô hình tự động học và nhận diện các đặc trưng quan trọng như nét chữ, hình dáng ký tự,

các dấu thanh, và các yếu tố cấu trúc khác. Khối này giúp mô hình có thể nhận diện các ký tự một cách chính xác, bất kể sự thay đổi trong cách viết hay độ nhiễu trong dữ liệu, đồng thời giảm thiểu sự phụ thuộc vào các đặc trưng được thiết kế thủ công.

Khối RNN (LSTM Block), đặc biệt là LSTM hai chiều, có vai trò xây dựng ngữ cảnh chuỗi từ các đặc trưng đã được trích xuất bởi khối chập. Trong bài toán nhận dạng chữ viết tay, các ký tự không thể được nhận diện một cách riêng lẻ mà cần phải hiểu mối quan hệ giữa các ký tự trong một chuỗi. Khối LSTM hai chiều giúp mô hình hiểu được ngữ cảnh của chuỗi ký tự bằng cách xử lý thông tin từ cả hai hướng: từ trái sang phải và từ phải sang trái. Điều này giúp cải thiện khả năng nhận dạng các từ hoặc đoạn văn bản dài, khi mà các ký tự cần phải được nhận diện trong ngữ cảnh tổng thể.

Hàm mất mát CTC (Connectionist Temporal Classification) đóng vai trò quan trọng trong việc giải quyết vấn đề độ dài không đồng nhất giữa đầu vào và đầu ra trong bài toán OCR. Với các ảnh chữ viết tay, chiều dài của chuỗi ký tự đầu ra có thể thay đổi so với độ dài chuỗi đầu vào. CTC Loss cho phép mô hình ánh xạ các đặc trưng đầu vào (từ các khối CNN và LSTM) vào chuỗi ký tự đầu ra mà không cần phải căn chỉnh chính xác độ dài của chúng. Điều này giúp mô hình có thể xử lý các khoảng trống, ký tự thiếu hoặc dư thừa, và đảm bảo dự đoán cuối cùng phù hợp với chuỗi ký tự thực tế mà không cần phải định dạng đầu ra một cách cứng nhắc, từ đó cải thiện hiệu quả nhận dạng chữ viết tay.

1.2. Ý tưởng chính

Tăng cường hiểu ngữ cảnh của chuỗi đầu ra thông qua LSTM hai chiều là một trong những yếu tố quan trọng giúp mô hình OCR cải thiện khả năng nhận diện chữ viết tay. Trong khi các mạng LSTM truyền thống chỉ xử lý thông tin theo một hướng, LSTM hai chiều (Bidirectional LSTM) cho phép mô hình học từ cả hai chiều: từ trái sang phải và ngược lại. Điều này rất quan trọng trong việc nhận dạng chữ viết tay, đặc biệt là trong tiếng Việt, nơi mà các ký tự có thể thay đổi nghĩa tuỳ thuộc vào ngữ cảnh của các ký tự trước và sau chúng, bao gồm cả dấu thanh. Việc hiểu ngữ cảnh từ cả hai chiều giúp mô hình có thể nhận diện chính xác các ký tự, dấu thanh và chữ ghép, giảm thiểu các sai sót trong nhận dạng, đặc biệt đối với các cụm từ hay từ ngữ phức tạp.

Sử dụng CTC Loss để giải quyết bài toán đầu ra có độ dài biến đổi là một yếu tố then chốt khác trong mô hình OCR. Trong bài toán nhận dạng chữ viết tay, độ dài của chuỗi ký tự

đầu ra thường không khớp với số lượng bước thời gian trong đầu vào (hình ảnh), do đó việc ánh xạ giữa đầu vào và đầu ra có thể gặp khó khăn. CTC Loss là một phương pháp hiệu quả giúp giải quyết vấn đề này, bằng cách cho phép mô hình tự động điều chỉnh độ dài của chuỗi đầu ra mà không cần phải căn chỉnh chính xác với từng ký tự. Hàm mất mát này giúp mô hình bỏ qua các ký tự dư thừa trong đầu ra và xử lý các khoảng trống giữa các ký tự, giúp cho việc nhận dạng chuỗi ký tự trở nên chính xác hơn, ngay cả khi đầu vào không được căn chỉnh hoàn hảo với đầu ra. Điều này rất hữu ích trong việc nhận diện các văn bản viết tay có sự biến đổi về khoảng cách và độ dài giữa các ký tự.

CHƯƠNG 2: DỮ LIỆU VÀ TIỀN XỬ LÝ DỮ LIỆU

2.1. Nguồn dữ liệu

Dữ liệu sử dụng trong bài toán nhận dạng chữ viết tay đến từ một bộ sưu tập các hình ảnh chứa văn bản viết tay, cùng với các nhãn tương ứng của chúng. Các hình ảnh này được lưu trữ trong thư mục gốc và bao gồm các tệp hình ảnh với nhiều định dạng khác nhau (như .jpg, .png, .bmp). Các nhãn của hình ảnh chứa văn bản viết tay được lưu trữ trong một tệp JSON, có tên là labels.json, với cấu trúc ánh xạ giữa đường dẫn hình ảnh và nhãn văn bản. Mỗi nhãn trong tệp JSON này mô tả văn bản viết tay trong hình ảnh tương ứng dưới dạng chuỗi ký tự.

Cấu trúc dữ liệu được thiết kế để dễ dàng đọc và xử lý thông qua Python, với các thư viện như pathlib và os để duyệt qua các thư mục chứa hình ảnh và nhãn. Việc ánh xạ giữa hình ảnh và nhãn được thực hiện thông qua một cấu trúc dữ liệu dictionary trong Python, với mỗi khóa là đường dẫn tới tệp hình ảnh và giá trị là chuỗi nhãn tương ứng. Dữ liệu này sẽ được sử dụng để huấn luyện mô hình nhận dạng chữ viết tay, với mỗi hình ảnh được gắn nhãn để mô hình có thể học cách chuyển đổi từ hình ảnh đầu vào thành chuỗi ký tự.

2.2. Phân tích dữ liệu ban đầu

Kích thước hình ảnh trong bộ dữ liệu đầu vào không đồng đều, điều này là một thách thức trong việc xử lý và huấn luyện mô hình. Để hiểu rõ hơn về đặc điểm của bộ dữ liệu, quá trình duyệt qua từng tệp hình ảnh đã được thực hiện để xác định chiều cao và chiều rộng tối thiểu và tối đa của các hình ảnh. Kết quả là chiều cao của hình ảnh dao động từ giá trị *min_height* đến *max_height*, trong khi chiều rộng dao động từ *min_width* đến *max_width*. Việc xác định những thông số này giúp chúng ta hiểu được sự phân bố kích thước của hình ảnh và là cơ sở để thực hiện các bước tiền xử lý tiếp theo, như điều chỉnh kích thước hình ảnh.

Bên cạnh việc phân tích kích thước hình ảnh, độ dài nhãn (số lượng ký tự trong văn bản) cũng là một yếu tố quan trọng trong quá trình chuẩn bị dữ liệu. Các nhãn trong bộ dữ liệu có độ dài khác nhau, với giá trị lớn nhất được sử dụng để xác định *max_label_len*. Việc xác định *max_label_len* là cần thiết để chuẩn hóa đầu ra của mô hình, đảm bảo rằng tất cả các

chuỗi nhãn sẽ có độ dài đồng nhất khi đưa vào mạng, dù cho độ dài thực tế của từng nhãn có thể khác nhau. Điều này sẽ giúp mô hình dễ dàng xử lý dữ liệu và cải thiện hiệu quả học tập.

2.3. Chia tập dữ liệu

Để huấn luyện và đánh giá mô hình một cách chính xác, việc chia bộ dữ liệu thành các tập huấn luyện và kiểm tra là rất quan trọng. Trong quá trình này, dữ liệu hình ảnh được phân chia thành hai phần: 80% được sử dụng cho việc huấn luyện mô hình, và 20% còn lại dành cho việc kiểm tra (validation). Việc phân chia này giúp đảm bảo rằng mô hình không bị quá khớp (overfitting) và có thể đánh giá được hiệu quả thực tế khi đối mặt với dữ liệu chưa thấy trước đó.

Để tăng tính ngẫu nhiên và đảm bảo tính nhất quán trong các lần chia dữ liệu, phương thức train_test_split của thư viện Scikit-learn đã được sử dụng. Tham số random_state=42 được đặt cố định, nhằm đảm bảo mỗi lần chạy chương trình sẽ có kết quả chia tập dữ liệu giống nhau, tạo điều kiện thuận lợi cho việc tái tạo kết quả và so sánh giữa các mô hình khác nhau. Sự phân chia này tạo ra một tập huấn luyện đủ lớn để mô hình học được các đặc trưng từ dữ liệu, đồng thời một tập kiểm tra với dữ liệu chưa thấy sẽ giúp đánh giá hiệu suất của mô hình trong thực tế.

2.4. Tiền xử lý dữ liệu

2.4.1. Mục tiêu

Mục tiêu chính của bước tiền xử lý dữ liệu là chuẩn hóa hình ảnh và nhãn sao cho phù hợp với yêu cầu của mô hình, giúp quá trình huấn luyện diễn ra hiệu quả hơn. Các hình ảnh chữ viết tay ban đầu có kích thước không đồng nhất, độ sáng tối khác nhau và có thể chứa nhiều yếu tố gây nhiễu. Do đó, việc xử lý và chuẩn hóa dữ liệu là rất quan trọng để đảm bảo chất lượng và tính chính xác trong quá trình nhận dạng.

Đối với hình ảnh, các bước tiền xử lý bao gồm chuyển đổi ảnh sang dạng ảnh xám để giảm thiểu số kênh màu không cần thiết, sau đó thay đổi kích thước hình ảnh về chiều cao cố định (118px) và điều chỉnh chiều rộng sao cho giữ tỷ lệ khung hình của ảnh. Tiếp theo, hình ảnh được bổ sung padding ở bên phải để đạt được chiều rộng cố định là 2167px, nhằm đảm bảo rằng tất cả hình ảnh đầu vào có kích thước đồng nhất. Thêm vào đó, một số kỹ thuật khác như làm mờ ảnh bằng bộ lọc Gaussian và ngưỡng hóa ảnh bằng phương pháp thích ứng (adaptive thresholding) giúp tăng cường độ tương phản của hình ảnh và làm nổi bật các ký tự, giảm bớt nhiễu không mong muốn. Cuối cùng, hình ảnh được chuẩn hóa bằng cách chia cho

255, giúp giá trị pixel nằm trong phạm vi [0, 1], phù hợp với yêu cầu đầu vào của mạng nơ-ron.

Đối với nhãn, mỗi ký tự trong chuỗi nhãn được mã hóa dưới dạng số nguyên, sau đó tất cả các nhãn được chuẩn hóa về chiều dài tối đa (max_label_len), sử dụng phương pháp padding để thêm giá trị 0 vào cuối chuỗi nhãn nếu chiều dài thực tế nhỏ hơn giá trị tối đa. Điều này đảm bảo rằng tất cả các nhãn có cùng độ dài, giúp mô hình dễ dàng xử lý trong quá trình huấn luyện và dự đoán.

2.4.2. Các bước xử lý hình ảnh

Để đảm bảo rằng hình ảnh đầu vào có thể được sử dụng hiệu quả trong mô hình nhận dạng chữ viết tay, cần thực hiện một số bước tiền xử lý cụ thể. Mỗi bước giúp tối ưu hóa chất lượng hình ảnh và chuẩn hóa kích thước để phù hợp với yêu cầu của mô hình mạng nơ-ron tích chập (CNN).

1. **Chuyển ảnh sang thang độ xám:** Bước đầu tiên là chuyển đổi tất cả các ảnh màu sang ảnh xám (grayscale). Hình ảnh màu thường có ba kênh màu (RGB), tuy nhiên trong bài toán nhận dạng chữ viết tay, chỉ cần thông tin về độ sáng của ảnh mà không cần phải xử lý các thông tin màu sắc. Việc chuyển sang ảnh xám giúp giảm độ phức tạp của dữ liệu và làm giảm chi phí tính toán mà không làm mất đi các đặc trưng cần thiết cho việc nhận dạng.

2. **Resize ảnh:** Sau khi chuyển ảnh sang thang độ xám, bước tiếp theo là điều chỉnh kích thước của ảnh. Chiều cao của tất cả ảnh được cố định về 118 pixel, đảm bảo tính đồng nhất trong kích thước hình ảnh. Để giữ tỷ lệ khung hình gốc, chiều rộng của ảnh được điều chỉnh sao cho phù hợp với chiều cao mới. Nếu chiều rộng của ảnh sau khi điều chỉnh nhỏ hơn 2167 pixel, ảnh sẽ được thêm padding ở bên phải. Padding được thực hiện bằng phương pháp median, giúp bảo toàn cấu trúc của ký tự và không làm mất thông tin quan trọng.

3. **Làm mờ ảnh:** Bước làm mờ ảnh được thực hiện với bộ lọc Gaussian. Phương pháp này giúp làm mượt các chi tiết không cần thiết và giảm bớt nhiễu, từ đó cải thiện chất lượng của hình ảnh trước khi tiếp tục với các bước xử lý tiếp theo. Việc giảm nhiễu là rất quan trọng trong các bài toán OCR, vì nhiễu có thể gây khó khăn cho việc nhận dạng ký tự.

4. **Ngưỡng hóa ảnh:** Sau khi làm mờ, ảnh sẽ được ngưỡng hóa bằng phương pháp thích ứng (adaptive thresholding) với thuật toán Gaussian. Phương pháp này chuyển ảnh sang dạng nhị phân (black and white), giúp tăng cường độ tương phản giữa chữ viết và

nền. Việc này sẽ làm cho ký tự trở nên nổi bật hơn, dễ dàng nhận diện hơn khi mạng nơ-ron thực hiện quá trình huấn luyện và dự đoán.

5. **Chuẩn hóa:** Để đảm bảo rằng tất cả các giá trị pixel của ảnh đều nằm trong phạm vi thích hợp cho việc huấn luyện, mỗi pixel của ảnh sẽ được chia cho 255. Sau bước này, giá trị pixel sẽ nằm trong khoảng [0, 1], giúp mô hình mạng nơ-ron có thể xử lý ảnh hiệu quả hơn, tránh các vấn đề phát sinh do sự khác biệt về độ sáng và mức độ tương phản.

6. **Mở rộng kênh:** Cuối cùng, một kênh mới sẽ được thêm vào ảnh để phù hợp với yêu cầu của mạng nơ-ron tích chập (CNN). Thay vì chỉ có một ma trận 2D đại diện cho ảnh, bây giờ ảnh sẽ có định dạng (118, 2167, 1), với '1' đại diện cho kênh đơn sắc. Việc mở rộng kênh này giúp mô hình có thể tiếp nhận dữ liệu theo định dạng mà các lớp CNN yêu cầu.

2.4.3. Xử lý nhãn

Quá trình xử lý nhãn là một bước quan trọng trong tiền xử lý dữ liệu cho bài toán nhận dạng chữ viết tay. Mỗi nhãn đại diện cho chuỗi ký tự cần nhận dạng từ hình ảnh đầu vào. Để mô hình có thể học và dự đoán chính xác, các nhãn cần được chuẩn hóa và chuyển đổi thành dạng mà mạng nơ-ron có thể hiểu và xử lý.

1. **Mã hóa nhãn (encode_to_labels) thành các giá trị số:** Trong bài toán OCR, mỗi ký tự trong nhãn sẽ được chuyển thành một giá trị số duy nhất. Quá trình này giúp biến các chuỗi ký tự thành các giá trị mà mô hình có thể sử dụng. Các ký tự như chữ cái, dấu thanh, hay khoảng trắng được ánh xạ tới các giá trị số nguyên thông qua một bảng mã hóa. Việc mã hóa này giúp loại bỏ sự phức tạp của các ký tự và giúp mô hình học từ các mẫu dữ liệu một cách hiệu quả hơn.

2. **Dùng pad_sequences để bổ sung số 0 vào cuối chuỗi:** Một đặc điểm quan trọng trong OCR là độ dài của các chuỗi nhãn có thể thay đổi giữa các mẫu dữ liệu. Để khắc phục vấn đề này và đảm bảo rằng tất cả các chuỗi nhãn có cùng độ dài, chúng ta sử dụng phương pháp padding. Trong đó, pad_sequences từ thư viện Keras được sử dụng để bổ sung các giá trị 0 vào cuối các chuỗi nhãn, sao cho độ dài của tất cả các nhãn đều bằng 240 (tương ứng với TIME_STEPS). Việc này giúp đảm bảo rằng các nhãn có độ dài đồng nhất, đồng thời cũng giúp mô hình dễ dàng học và dự đoán các chuỗi ký tự với độ dài cố định.

2.5. Kết quả của tiền xử lý

Sau khi thực hiện các bước tiền xử lý dữ liệu, chúng ta đã thu được một tập hợp dữ liệu đã được chuẩn hóa, sẵn sàng để đưa vào mô hình huấn luyện. Các kết quả sau tiền xử lý được phân loại thành các thành phần chính như sau:

- **training_img:** Đây là danh sách các hình ảnh đầu vào đã được chuẩn hóa. Sau các bước xử lý, các hình ảnh này đã được chuyển sang dạng thang độ xám, điều chỉnh kích thước đồng nhất, làm mờ và ngưỡng hóa để tăng cường độ tương phản. Các hình ảnh này đã được chuẩn bị sẵn sàng với các giá trị pixel trong khoảng [0, 1], giúp mô hình học hiệu quả hơn.
- **training_txt:** Các nhãn (chuỗi ký tự) đã được mã hóa thành các giá trị số và bổ sung padding để đảm bảo rằng độ dài của tất cả các nhãn đều giống nhau, bằng 240 bước (tương ứng với TIME_STEPS). Các giá trị số này là đầu vào cho mạng nơ-ron trong quá trình huấn luyện.
- **train_input_length:** Đây là độ dài cố định của chuỗi đầu vào, đảm bảo rằng tất cả các hình ảnh đầu vào có số bước thời gian (time steps) bằng nhau. Với thiết lập TIME_STEPS = 240, các hình ảnh sẽ được chuẩn hóa về độ dài đầu vào đồng nhất, giúp mô hình có thể xử lý hiệu quả mà không bị ảnh hưởng bởi sự thay đổi kích thước của hình ảnh gốc.
- **train_label_length:** Đây là độ dài thực tế của các nhãn trước khi bổ sung padding. Mỗi nhãn có thể có độ dài khác nhau, và giá trị này phản ánh số lượng ký tự thực tế có trong mỗi nhãn. Việc tính toán độ dài thực tế của nhãn giúp mô hình hiểu được sự biến đổi trong số lượng ký tự, hỗ trợ quá trình huấn luyện và đánh giá.
- **orig_txt:** Đây là các nhãn gốc, chứa chuỗi ký tự ban đầu, chưa qua mã hóa và padding. Các nhãn gốc này sẽ được sử dụng để tham chiếu trong quá trình đánh giá mô hình, giúp so sánh kết quả dự đoán của mô hình với chuỗi ký tự thực tế.

2.6. Ý nghĩa của tiền xử lý

Tiền xử lý dữ liệu đóng một vai trò quan trọng trong việc chuẩn bị dữ liệu cho mô hình học sâu, đặc biệt là trong bài toán nhận dạng chữ viết tay. Các bước tiền xử lý không chỉ giúp chuẩn hóa dữ liệu mà còn nâng cao hiệu quả học của mô hình, đảm bảo tính tương thích giữa các thành phần trong hệ thống.

Đầu tiên, việc đồng nhất hóa kích thước hình ảnh và nhãn là một trong những yếu tố quan trọng trong tiền xử lý. Quá trình điều chỉnh kích thước hình ảnh và chuẩn hóa độ dài

nhãn giúp tạo ra một đầu vào đồng nhất cho mô hình, giảm thiểu sự không đồng nhất giữa các mẫu dữ liệu. Điều này không chỉ giúp mô hình học hiệu quả hơn mà còn tránh các lỗi do sự khác biệt trong kích thước hình ảnh hoặc độ dài nhãn gây ra.

Thêm vào đó, các bước như làm mờ và ngưỡng hóa ảnh đã giúp mô hình tập trung vào các đặc trưng quan trọng của hình ảnh, như nét chữ và các dấu thanh, đồng thời loại bỏ những yếu tố không cần thiết. Quá trình làm mờ bằng bộ lọc Gaussian giúp giảm nhiễu, trong khi ngưỡng hóa ảnh với thuật toán Gaussian giúp tăng cường độ tương phản giữa chữ viết và nền. Điều này giúp mô hình dễ dàng nhận diện các ký tự trong hình ảnh, cải thiện khả năng nhận dạng chính xác.

Cuối cùng, các bước như padding và mã hóa chuỗi nhãn cũng đóng vai trò quan trọng trong việc đảm bảo tính tương thích giữa các phần của mô hình. Việc sử dụng padding giúp chuẩn hóa độ dài của các chuỗi nhãn, trong khi mã hóa chuỗi giúp chuyển đổi các ký tự thành dạng số mà mô hình có thể hiểu và xử lý. Điều này giúp mô hình học tốt hơn từ dữ liệu chữ viết tay OCR, đồng thời đảm bảo rằng mô hình có thể xử lý được các trường hợp đầu vào không đồng nhất, từ đó cải thiện độ chính xác và hiệu quả của quá trình huấn luyện.

CHƯƠNG 3: HUẤN LUYỆN VÀ TRIỂN KHAI

3.1. Quy trình huấn luyện

3.1.1. Thông số chính trong quá trình huấn luyện

Trong quy trình huấn luyện mô hình nhận dạng chữ viết tay, việc chọn lựa các thông số huấn luyện phù hợp là rất quan trọng để đảm bảo mô hình đạt hiệu quả cao nhất. Các thông số chính trong quá trình huấn luyện bao gồm batch size, epochs và learning rate, mỗi thông số đóng một vai trò quan trọng trong việc điều chỉnh quá trình học của mô hình.

Batch size, hay kích thước mỗi lô dữ liệu, là số lượng mẫu được đưa vào mô hình trong mỗi lần cập nhật trọng số. Việc lựa chọn giá trị phù hợp cho batch size sẽ giúp cân bằng giữa tốc độ huấn luyện và mức sử dụng bộ nhớ. Batch size quá nhỏ có thể làm cho quá trình huấn luyện chậm hơn, trong khi quá lớn lại tiêu tốn nhiều bộ nhớ và có thể làm giảm tính chính xác của mô hình do không đủ tính ngẫu nhiên trong việc học.

Số lượng epochs, hay số lần lặp lại mô hình trên toàn bộ tập dữ liệu huấn luyện, quyết định mức độ "sâu" mà mô hình học. Mô hình được huấn luyện nhiều epochs có thể học được các đặc trưng phức tạp hơn từ dữ liệu, nhưng nếu quá nhiều epochs sẽ dẫn đến nguy cơ quá khớp (overfitting), khi mô hình học quá nhiều chi tiết của dữ liệu huấn luyện và không tổng quát được với dữ liệu mới. Vì vậy, cần phải chọn số lượng epochs phù hợp để mô hình học hiệu quả mà không bị overfitting.

Learning rate, hay tốc độ học, là thông số quyết định mức độ thay đổi của trọng số trong mỗi bước cập nhật. Một learning rate nhỏ sẽ giúp mô hình hội tụ ổn định nhưng có thể mất thời gian lâu hơn, trong khi một learning rate quá lớn có thể giúp mô hình học nhanh hơn nhưng dễ dẫn đến dao động và không hội tụ được. Việc lựa chọn learning rate phù hợp giúp mô hình học nhanh mà vẫn đảm bảo được độ chính xác trong quá trình huấn luyện.

3.1.2. Chi tiết thuật toán tối ưu (Adam)

Trong quá trình huấn luyện mô hình, việc chọn lựa thuật toán tối ưu phù hợp là một yếu tố quan trọng để nâng cao hiệu quả học của mô hình. Một trong những thuật toán tối ưu phổ biến và hiệu quả nhất hiện nay là Adam (Adaptive Moment Estimation), được sử dụng rộng rãi trong các bài toán học sâu, đặc biệt là các mô hình phức tạp như OCR.

Adam kết hợp hai yếu tố quan trọng: động lượng (momentum) và các điều chỉnh thích ứng của learning rate. Động lượng giúp tăng tốc quá trình học bằng cách duy trì hướng đi của

gradient từ các bước cập nhật trước, giúp mô hình tránh được sự dao động mạnh trong quá trình huấn luyện. Đồng thời, Adam tự động điều chỉnh learning rate cho từng tham số riêng biệt, dựa trên các giá trị trung bình và phương sai của gradient từ các bước huấn luyện trước. Điều này giúp các tham số có gradient lớn học nhanh hơn, trong khi các tham số có gradient nhỏ sẽ học chậm hơn, từ đó đạt được sự tối ưu hóa hiệu quả hơn trong toàn bộ mạng.

Ưu điểm lớn của Adam là khả năng tự điều chỉnh learning rate, điều này rất quan trọng khi làm việc với các dữ liệu phức tạp và kiến trúc mạng phức tạp như OCR chữ viết tay. Thêm vào đó, Adam giúp giảm thiểu nguy cơ mô hình bị rơi vào cực tiểu cục bộ, một vấn đề phổ biến khi sử dụng các thuật toán tối ưu đơn giản như Gradient Descent. Với khả năng học nhanh và ổn định, Adam là một sự lựa chọn lý tưởng để huấn luyện các mô hình nhận dạng chữ viết tay, đặc biệt là trong các bài toán OCR yêu cầu độ chính xác cao và xử lý dữ liệu không đồng nhất.

3.2. Hàm mất mát

3.2.1. Giải thích CTC Loss (Connectionist Temporal Classification)

Hàm mất mát CTC (Connectionist Temporal Classification) là một phương pháp tối ưu quan trọng trong các bài toán nhận dạng chuỗi không đồng nhất, như OCR (Optical Character Recognition) và nhận dạng giọng nói. CTC được thiết kế để giải quyết vấn đề đặc thù của các bài toán này, nơi độ dài của đầu vào và đầu ra không khớp nhau.

Trong mô hình OCR, chuỗi đặc trưng đầu vào từ các lớp Convolutional Neural Network (CNN) có chiều dài cố định (ví dụ 240 bước), trong khi chuỗi đầu ra, tức là các ký tự trong văn bản, có độ dài thay đổi tùy vào nội dung văn bản. Điều này tạo ra một sự không khớp giữa đầu vào và đầu ra, mà các phương pháp học sâu truyền thống không thể xử lý trực tiếp. Để giải quyết vấn đề này, CTC cung cấp một cơ chế cho phép mô hình ánh xạ giữa chuỗi đầu vào và đầu ra mà không cần căn chỉnh thủ công. CTC có thể xử lý sự không khớp này bằng cách đưa vào các khoảng trống (blanks) vào quá trình dự đoán, giúp mô hình linh hoạt trong việc xác định đâu là điểm bắt đầu và kết thúc của một ký tự, cũng như tự động xử lý các khoảng trống giữa các ký tự.

Cụ thể, CTC cho phép mô hình dự đoán một chuỗi các ký tự với độ dài thay đổi từ chuỗi đặc trưng có độ dài cố định, đồng thời giảm thiểu sự cần thiết phải căn chỉnh chuỗi đầu vào và đầu ra một cách chính xác. Hàm mất mát CTC tính toán xác suất của một chuỗi đầu ra dựa trên các bước dự đoán từ đầu vào, và điều chỉnh mô hình để tối thiểu hóa sự khác biệt giữa đầu ra dự đoán và nhãn thực tế. Đây là một điểm mạnh đặc biệt trong các ứng dụng OCR, nơi sự không đồng nhất về độ dài giữa đặc trưng và chuỗi ký tự là một vấn đề phổ biến.

3.2.2. Lý do sử dụng CTC loss

Trong bài toán nhận dạng chữ viết tay, một trong những thách thức lớn nhất là sự không đồng nhất giữa hình ảnh đầu vào và chuỗi ký tự đầu ra. Cụ thể, ảnh chữ viết tay thường không có sự căn chỉnh chính xác giữa các yếu tố trong ảnh và nhãn văn bản (như các ký tự và dấu thanh). Ví dụ, trong tiếng Việt, dấu thanh có thể đặt ở vị trí khác nhau hoặc không rõ ràng trong hình ảnh, điều này làm cho việc căn chỉnh trực tiếp các bước thời gian giữa đặc trưng đầu vào và ký tự trong nhãn trở nên rất khó khăn.

Hàm mất mát CTC là giải pháp lý tưởng trong trường hợp này, vì nó không yêu cầu sự căn chỉnh chính xác giữa các bước thời gian trong chuỗi đầu vào và chuỗi đầu ra. Thay vì yêu cầu mỗi bước của mô hình phải dự đoán một ký tự cụ thể, CTC cho phép mô hình dự đoán một chuỗi các ký tự mà không cần phải gán nhãn cho mỗi bước thời gian. Điều này giúp mô hình tự động xử lý các khoảng trống giữa các ký tự hoặc phần dữ liệu dư thừa (như các khoảng trắng giữa chữ cái hoặc dấu thanh), mà không cần phải căn chỉnh chặt chẽ giữa từng bước và nhãn cụ thể.

Với cách tiếp cận này, CTC giúp mô hình học tốt hơn từ dữ liệu không được căn chỉnh chính xác, đồng thời làm giảm sự phụ thuộc vào việc phải gán nhãn thủ công cho từng bước trong chuỗi đặc trưng. Đây là lý do vì sao CTC được sử dụng rộng rãi trong các hệ thống OCR cho chữ viết tay, đặc biệt là khi xử lý các ngôn ngữ có tính phức tạp như tiếng Việt.

3.2.3. Vai trò của input_length và label_length

Trong quá trình huấn luyện mô hình OCR sử dụng CTC loss, hai tham số quan trọng là input_length và label_length đóng vai trò quyết định trong việc xác định cách ánh xạ giữa chuỗi đặc trưng đầu vào và chuỗi ký tự đầu ra.

input_length biểu thị chiều dài của chuỗi đặc trưng đầu vào, được trích xuất từ các lớp CNN sau khi xử lý hình ảnh. Đây là số lượng bước thời gian (timesteps) trong chuỗi đặc trưng mà mô hình sẽ sử dụng để học các thông tin từ ảnh. Do việc xử lý dữ liệu hình ảnh thường dẫn đến sự thay đổi về chiều rộng của ảnh (sau các bước resize và padding), input_length sẽ giúp xác định số lượng bước thời gian thực tế mà mô hình cần học từ hình ảnh đó.

label_length, ngược lại, là chiều dài thực tế của chuỗi nhãn, tức là số lượng ký tự trong chuỗi đầu ra (nhãn văn bản). Vì độ dài của chuỗi ký tự có thể khác nhau đối với mỗi ví dụ trong tập huấn luyện, label_length cho phép mô hình hiểu rõ số lượng ký tự trong chuỗi kết quả cần dự đoán. Đối với các bài toán OCR, chiều dài chuỗi nhãn có thể thay đổi tùy vào nội dung của văn bản viết tay, do đó việc xác định label_length là rất quan trọng để mô hình có thể điều chỉnh và học chính xác.

Cả hai tham số này giúp hàm mất mát CTC xác định rõ ràng vị trí cần ánh xạ giữa đầu vào (chuỗi đặc trưng từ CNN) và đầu ra (chuỗi ký tự), giúp mô hình học một cách linh hoạt và hiệu quả hơn. Việc điều chỉnh độ dài của chuỗi đầu vào và chuỗi đầu ra một cách chính xác đảm bảo rằng mô hình có thể tối ưu hóa quá trình học mà không gặp phải sự mất đồng bộ giữa các bước thời gian và nhãn văn bản.

3.3. Đánh giá mô hình

Để đánh giá hiệu quả của mô hình OCR, một tập kiểm thử (validation set) được sử dụng, được tách ra từ dữ liệu ban đầu và chiếm 20% tổng số hình ảnh. Dữ liệu kiểm thử này không tham gia vào quá trình huấn luyện mà chỉ được sử dụng để kiểm tra khả năng tổng quát hóa của mô hình. Việc này giúp đảm bảo rằng mô hình không chỉ hoạt động tốt trên tập huấn luyện mà còn có khả năng nhận dạng chính xác các ký tự từ các dữ liệu chưa thấy.

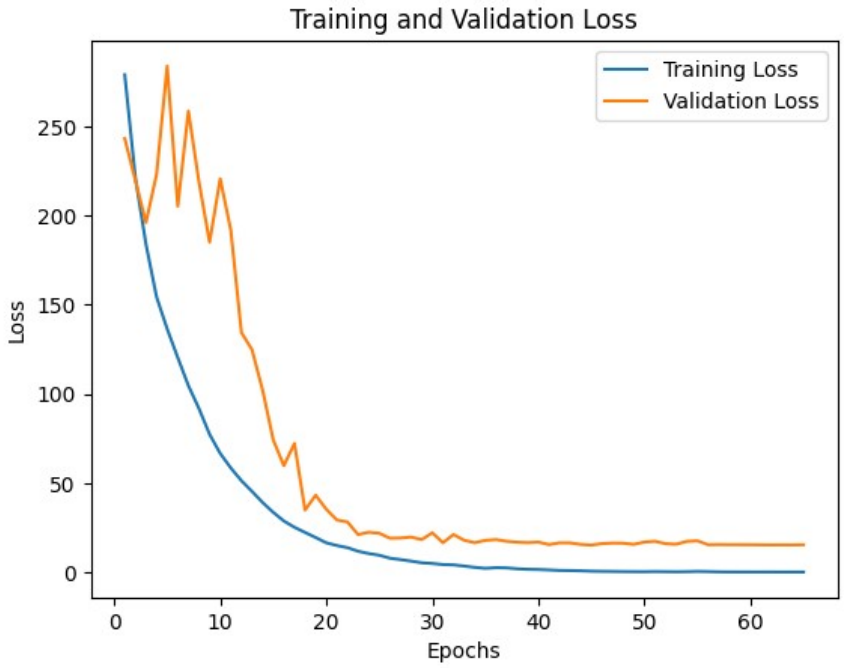
Các chỉ số đánh giá hiệu quả mô hình bao gồm độ chính xác (Accuracy) và tỷ lệ nhận dạng sai (Error Rate). Độ chính xác được tính bằng tỷ lệ giữa số chuỗi ký tự được dự đoán đúng hoàn toàn và tổng số chuỗi trong tập kiểm thử. Đây là một chỉ số quan trọng phản ánh mức độ chính xác của mô hình trong việc nhận dạng chữ viết tay. Tỷ lệ nhận dạng sai, ngược lại, đo lường mức độ sai lệch của mô hình trong việc dự đoán ký tự hoặc chuỗi ký tự. Tỷ lệ này bao gồm các chỉ số như CER (Character Error Rate) và WER (Word Error Rate). CER đo lường sai lệch giữa ký tự dự đoán và nhãn thực tế, trong khi WER đánh giá sai lệch theo từ, giúp hiểu rõ hơn về khả năng nhận dạng từ của mô hình.

Để đánh giá mô hình một cách chính xác, chúng ta sử dụng phương pháp CTC decode. Phương pháp này chuyển đổi chuỗi đặc trưng đầu ra từ mô hình thành chuỗi ký tự có thể đọc được. Sau đó, chúng ta so sánh chuỗi ký tự đã giải mã với nhãn thực tế để tính toán các chỉ số đánh giá như độ chính xác và các tỷ lệ lỗi (CER và WER). Quá trình này giúp chúng ta đánh giá mức độ phù hợp của mô hình đối với dữ liệu thực tế và khả năng tổng quát hóa trong các tình huống mới.

CHƯƠNG 4: KẾT QUẢ

4.1. Hiệu năng

Kết quả của mô hình được đánh giá dựa trên ba chỉ số chính: **Character Error Rate (CER)**, **Word Error Rate (WER)**, và **Sequence Error Rate (SER)**. Mô hình đạt được **CER = 0.0476**, tương đương với tỷ lệ lỗi theo ký tự là 4.76%, cho thấy khả năng nhận dạng chính xác hầu hết các ký tự trong chuỗi. Chỉ số **WER = 0.1566**, tức tỷ lệ lỗi theo từ là 15.66%, phản ánh sự khác biệt nhỏ giữa các từ dự đoán và từ thực tế. Tuy nhiên, chỉ số **SER = 0.8098**, nghĩa là tỷ lệ chuỗi ký tự hoàn toàn sai là 80.98%, cho thấy mô hình còn khó khăn trong việc nhận dạng chính xác toàn bộ chuỗi đầu ra.



Các kết quả này được so sánh giữa tập huấn luyện và kiểm thử thông qua biểu đồ và bảng số liệu, giúp đánh giá khả năng tổng quát hóa của mô hình. Nhìn chung, hiệu năng cho thấy mô hình hoạt động tốt ở mức độ ký tự và từ, nhưng cần cải thiện để giảm lỗi toàn chuỗi, đặc biệt là trong các trường hợp chuỗi dài hoặc phức tạp.

4.2. Thảo luận

Mô hình có một số **điểm mạnh** đáng kể, bao gồm khả năng nhận dạng chính xác các ký tự đơn lẻ và các từ ngắn, nhờ kiến trúc kết hợp CNN và BiLSTM. Việc sử dụng hàm mất mát CTC cũng giúp giải quyết vấn đề độ dài không khớp giữa đầu vào

và đầu ra, đặc biệt trong OCR chữ viết tay. Chỉ số CER thấp phản ánh mô hình có thể học tốt các đặc trưng ký tự viết tay, ngay cả khi đối mặt với sự biến dạng hoặc các phong cách viết khác nhau.

Tuy nhiên, mô hình vẫn tồn tại **hạn chế**. Thứ nhất, chỉ số SER cao cho thấy khó khăn trong việc nhận dạng chính xác toàn chuỗi, đặc biệt trong các trường hợp có dấu thanh phức tạp, ký tự viết tay không đều, hoặc các từ dài. Ngoài ra, chữ viết tay tiếng Việt có nhiều đặc trưng phức tạp, như dấu thanh có thể xuất hiện tách rời hoặc ghi đè lên ký tự, dễ gây nhầm lẫn cho mô hình. Nhiễu trong hình ảnh hoặc sự không nhất quán về kích thước và nét chữ cũng là thách thức lớn, ảnh hưởng trực tiếp đến kết quả dự đoán.

Trong tương lai, cần tập trung vào việc cải thiện SER thông qua tối ưu hóa mô hình hoặc thử nghiệm các kiến trúc tiên tiến hơn như Transformer OCR. Đồng thời, việc bổ sung dữ liệu huấn luyện đa dạng hơn và cải thiện tiền xử lý, như giảm nhiễu hoặc cân bằng tỷ lệ giữa các loại mẫu trong tập dữ liệu, cũng có thể giúp tăng cường hiệu quả nhận dạng toàn chuỗi.

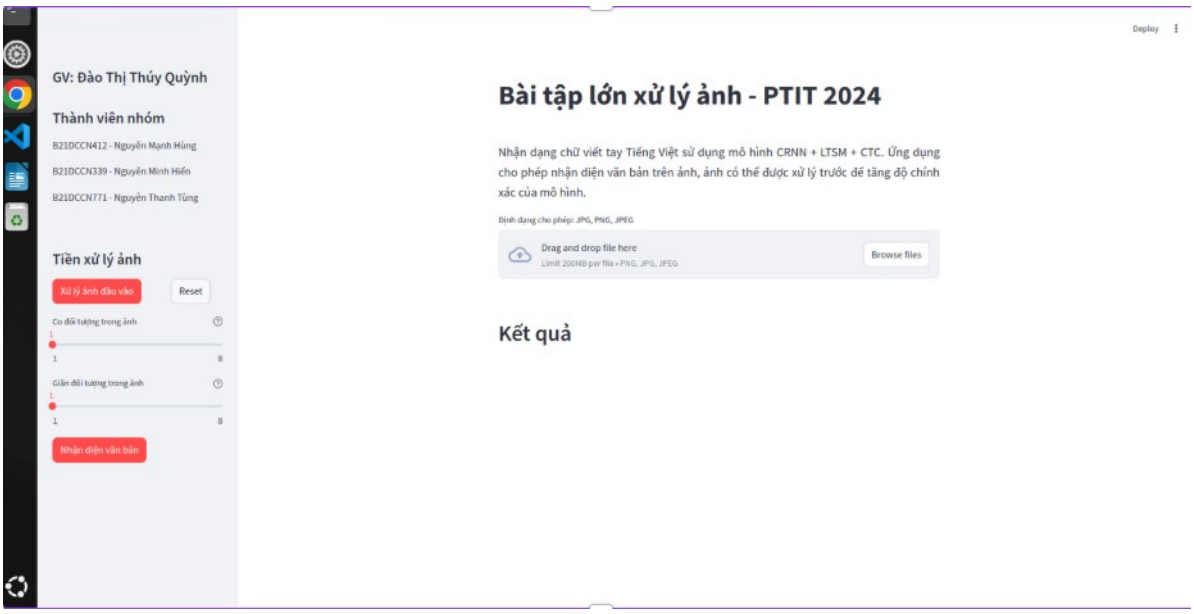
CHƯƠNG 5: XÂY DỰNG GIAO DIỆN

5.1. Mục tiêu xây dựng giao diện

Mục tiêu của việc xây dựng giao diện là tạo ra một công cụ đơn giản và dễ sử dụng, tận dụng thư viện **Streamlit** để hỗ trợ người dùng trong việc xử lý và nhận dạng chữ viết tay. Thư viện **Streamlit** được lựa chọn nhờ tính đơn giản, mạnh mẽ và thiết kế dành riêng cho các ứng dụng **machine learning** hoặc **data science**. Nó cho phép xây dựng giao diện tương tác nhanh chóng mà không đòi hỏi kinh nghiệm sâu về lập trình giao diện người dùng. Ứng dụng giao diện này sẽ giúp người dùng giao tiếp dễ dàng với mô hình OCR, chỉ cần tải ảnh lên và thao tác trực tiếp mà không cần hiểu biết chuyên sâu về các bước xử lý nội bộ.

Giao diện được thiết kế để đáp ứng đầy đủ các chức năng cần thiết, bao gồm hỗ trợ **tiền xử lý ảnh** như co dãn đối tượng và chuẩn bị dữ liệu đầu vào tối ưu cho mô hình. Ngoài ra, hệ thống tích hợp mô hình OCR đã huấn luyện, cho phép dự đoán nội dung chữ viết tay một cách chính xác. Kết quả nhận dạng sẽ được hiển thị trực quan trên giao diện, kèm tính năng sao chép văn bản để người dùng dễ dàng sử dụng.

Một yếu tố quan trọng khác là giao diện phải hỗ trợ tương tác trực quan, cung cấp kết quả nhanh chóng và thân thiện với người dùng. Người dùng có thể điều chỉnh các tham số tiền xử lý ảnh thông qua thanh trượt (**slider**) và quan sát trực tiếp sự thay đổi trên ảnh. Giao diện được tối ưu hóa để đảm bảo tốc độ xử lý nhanh nhưng vẫn duy trì độ chính xác cao, phù hợp cho các ứng dụng thực tế như quét tài liệu hay nhận dạng địa chỉ. Tất cả các thành phần đều được tổ chức gọn gàng và dễ hiểu, giúp người dùng sử dụng một cách dễ dàng và hiệu quả.



5.2. Công nghệ sử dụng

Streamlit được sử dụng làm framework chính để xây dựng giao diện web, nhờ khả năng tạo ứng dụng một cách nhanh chóng và hiệu quả. Với **Streamlit**, các thành phần giao diện như thanh trượt (**slider**), nút bấm (**button**), và khung hiển thị có thể được tạo ra dễ dàng mà không yêu cầu nhiều dòng mã, giúp tối ưu hóa thời gian phát triển.
Ngôn ngữ lập trình **Python** đóng vai trò trung tâm trong hệ thống. Python không chỉ được sử dụng để tích hợp các bước xử lý ảnh và mô hình OCR mà còn đảm nhiệm việc điều phối logic xử lý, giao tiếp giữa giao diện và các thành phần phía sau.

Một số thư viện hỗ trợ được sử dụng để hoàn thiện các chức năng của ứng dụng:

- **Pillow** hoặc **OpenCV**: Được dùng để thực hiện các bước tiền xử lý ảnh, bao gồm chuyển đổi thang độ xám, thay đổi kích thước, và làm mịn ảnh. Các thư viện này đảm bảo ảnh đầu vào được chuẩn hóa, tối ưu hóa để mô hình OCR hoạt động hiệu quả nhất.
- **TensorFlow** hoặc **Keras**: Hỗ trợ tải và sử dụng mô hình CRNN đã huấn luyện. Đồng thời, thư viện này cũng giúp giải mã đầu ra bằng **CTC** (Connectionist Temporal Classification), cho phép nhận dạng chính xác các chuỗi ký tự từ hình ảnh mà không yêu cầu căn chỉnh cụ thể.

Bộ công nghệ này kết hợp chặt chẽ với nhau, đảm bảo giao diện không chỉ trực quan mà còn mang lại hiệu suất cao, đáp ứng đầy đủ các yêu cầu của bài toán nhận dạng chữ viết tay tiếng Việt.

5.3. Quy trình xử lý trong giao diện

1. Nhập dữ liệu
Người dùng có thể tải ảnh lên thông qua nút **"Browse files"**. Hệ thống hỗ trợ các định dạng ảnh phổ biến như **JPG, PNG, JPEG** và giới hạn kích thước tệp tối đa là **200MB**. Sau khi ảnh được tải lên, nó sẽ hiển thị trên giao diện để người dùng dễ dàng kiểm tra.

2. Tiền xử lý ảnh
Giao diện cung cấp các tùy chọn chỉnh sửa ảnh trực quan thông qua thanh trượt (**slider**) nhằm tối ưu hóa ảnh đầu vào cho mô hình OCR:

- **Co đối tượng trong ảnh:** Tính năng này thay đổi tỷ lệ hình ảnh, giúp làm rõ nét chữ viết tay trong các ảnh có tỷ lệ không đồng nhất.
- **Giãn đối tượng trong ảnh:** Cho phép người dùng điều chỉnh kích thước hoặc độ dãn theo chiều ngang/dọc của các ký tự, đặc biệt hữu ích khi chữ viết tay bị bóp méo.
  Kết quả tiền xử lý được hiển thị ngay trên giao diện, cho phép người dùng quan sát và điều chỉnh các thông số phù hợp trước khi thực hiện nhận diện.

3. Nhận diện văn bản

Người dùng nhấn nút **"Nhận diện văn bản"** để bắt đầu quá trình OCR. Ứng dụng sẽ sử dụng mô hình đã huấn luyện để nhận diện nội dung chữ viết tay từ ảnh đầu vào. Kết quả nhận dạng sẽ được hiển thị ngay lập tức ở phần **"Kết quả"** trên giao diện.

4. Chức năng sao chép và reset

- **Copy and reset:** Nút này cho phép người dùng sao chép nội dung nhận dạng trực tiếp vào clipboard để sử dụng ngay, đồng thời làm mới giao diện để bắt đầu một phiên làm việc mới.
- **Reset:** Khi người dùng muốn làm mới hoàn toàn, nút này sẽ xóa toàn bộ dữ liệu ảnh và kết quả hiện tại, đưa giao diện trở về trạng thái ban đầu, sẵn sàng cho một quy trình xử lý mới.

5.4. Các thành phần giao diện chính

1. Header

- **Tiêu đề:** Giao diện có tiêu đề nổi bật là **"Bài tập lớn xử lý ảnh - PTIT 2024"**, được đặt ở phía trên cùng để thể hiện rõ chủ đề của ứng dụng.
- **Thông tin giảng viên và nhóm thực hiện:** Được hiển thị ở cột bên trái, bao gồm tên giảng viên hướng dẫn và danh sách thành viên nhóm, giúp dễ dàng xác định đội ngũ thực hiện.

2. Khu vực tải ảnh

- **Nút tải ảnh:** Giao diện hỗ trợ người dùng tải ảnh thông qua nút **"Browse files"**, với tính năng kéo thả hoặc chọn file từ thiết bị cá nhân.
- **Thông tin tệp:** Sau khi tải lên, thông tin chi tiết của file, như tên file và kích thước, sẽ được hiển thị ngay dưới nút tải ảnh để người dùng xác nhận đúng tệp dữ liệu.

3. Khu vực tiền xử lý

- **Thanh trượt điều chỉnh: Co đối tượng trong ảnh và Giãn đối tượng trong ảnh.**
- **Hiển thị ảnh:** Người dùng có thể xem ảnh gốc và ảnh sau khi tiền xử lý ở các trạng thái khác nhau (như ảnh nhị phân, làm mượt, căn chỉnh) để kiểm tra kết quả điều chỉnh.

4. Khu vực nhận diện văn bản

- **Hộp hiển thị kết quả:** Sau khi nhận diện, nội dung chữ viết tay được chuyển đổi thành văn bản và hiển thị trong hộp văn bản phía dưới. Hộp này hỗ trợ sao chép kết quả chỉ với một cú nhấp chuột.
- **Nút Nhận diện văn bản:** Nút này kích hoạt mô hình OCR để xử lý ảnh đã được tiền xử lý và trả về kết quả nhận dạng trong hộp văn bản.

5. Chức năng reset và sao chép

- **Nút Copy and reset:** Kết hợp chức năng sao chép nội dung nhận dạng vào clipboard và xóa toàn bộ dữ liệu để làm mới giao diện, giúp người dùng nhanh chóng chuẩn bị cho phiên làm việc mới.

- **Nút Reset:** Làm mới hoàn toàn giao diện, xóa mọi tệp đã tải lên, các tùy chỉnh và kết quả nhận dạng, đưa ứng dụng trở về trạng thái ban đầu.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Trong nghiên cứu này, mô hình OCR sử dụng học sâu (deep learning) đã đạt được những kết quả khả quan trong việc nhận dạng chữ viết tay tiếng Việt. Nhờ vào việc sử dụng kết hợp các kiến trúc mạng nơ-ron như CNN (Convolutional Neural Network) để trích xuất đặc trưng hình ảnh và LSTM (Long Short-Term Memory) để mô hình hóa chuỗi ký tự, mô hình đã có thể nhận dạng được chữ viết tay với độ chính xác cao. Việc sử dụng hàm mất mát CTC (Connectionist Temporal Classification) đã giúp xử lý hiệu quả sự khác biệt về độ dài giữa đầu vào và đầu ra, cải thiện độ chính xác của mô hình khi làm việc với các chuỗi ký tự có độ dài biến đổi. Phương pháp học sâu cho thấy sự vượt trội trong việc tự động học và cải thiện khả năng nhận dạng chữ viết tay, đặc biệt là khi dữ liệu có sự biến đổi lớn về kiểu chữ và dấu thanh, điều rất quan trọng trong ngữ cảnh tiếng Việt.

Mặc dù kết quả hiện tại là tích cực, mô hình vẫn còn có thể được tối ưu hóa thêm. Một trong những hướng phát triển quan trọng là cải thiện khả năng nhận dạng với các dữ liệu đa dạng hơn, chẳng hạn như chữ viết tay từ nhiều người viết khác nhau hoặc các ngữ cảnh khác nhau. Việc thử nghiệm với các mô hình tiên tiến hơn như Transformer hoặc các mô hình kết hợp giữa CNN và Attention Mechanism có thể giúp tăng cường hiệu quả nhận dạng. Bên cạnh đó, việc triển khai ứng dụng thực tế cho mô hình OCR này, chẳng hạn như phát triển ứng dụng nhận dạng chữ viết tay trên các thiết bị di động hoặc trong các hệ thống tự động hóa văn phòng, sẽ mở ra nhiều cơ hội ứng dụng thiết thực cho công nghệ nhận dạng văn bản, từ đó thúc đẩy sự phát triển mạnh mẽ của các hệ thống hỗ trợ nhận dạng chữ viết tay.

TÀI LIỆU THAM KHẢO

1) Graves, A., & Schmidhuber, J. (2009). *Offline handwriting recognition with multidimensional recurrent neural networks*. In Proceedings of the 22nd international conference on machine learning (ICML-09) (pp. 577-584).

2) Ba, J., & Caruana, R. (2014). *Do deep nets really need to be deep?* In Advances in neural information processing systems (NIPS 2014) (pp. 2654-2662).

3) Kingma, D. P., & Ba, J. (2015). *Adam: A method for stochastic optimization*. In Proceedings of the 3rd international conference on learning representations (ICLR 2015). Retrieved from https://arxiv.org/abs/1412.6980

4) Chollet, F. (2015). *Keras: The Python deep learning library*. Retrieved from https://github.com/keras-team/keras

5) Zhang, Z., & Wang, Y. (2019). *Connectionist temporal classification: Applications and techniques in speech recognition*. Journal of Machine Learning Research, 20(1), 1-37.

6) LeCun, Y., & Bengio, Y. (1995). *Convolutional networks for images, speech, and time series*. The Handbook of Brain Theory and Neural Networks, 255-258. MIT Press.

7) Yu, L., & Cho, K. (2016). *Learning for structured prediction*. Journal of Machine Learning Research, 17(1), 1119-1151.

8) Zhan, X., & Wang, Q. (2020). *A comprehensive review on deep learning in optical character recognition*. Pattern Recognition, 103, 107261.