NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Tp. Hồ Chí Minh, ngày ….. tháng ….. năm …..

Giáo viên hướng dẫn

NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Tp. Hồ Chí Minh, ngày ….. tháng ….. năm …..

Giáo viên phản biện

Khoa Công Nghệ Thông Tin
Bộ môn Khoa học Máy tính

ĐỀ CƯƠNG CHI TIẾT

Tên Đề Tài: Nhận diện chữ in trên Chứng minh nhân dân
Giáo viên hướng dẫn: Tiến sĩ Lê Mai Tùng Thạc sĩ Lê Ngọc Thành
Thời gian thực hiện: 15/01/2020 đến ngày 15/07/2020
Sinh viên thực hiện: Nguyễn Văn Tuấn – 1653100 Nguyễn Trọng An – 1653107
Loại đề tài: Nghiên cứu có ứng dụng demo

Nội Dung Đề Tài:

1. Xây dựng chương trình trích xuất thông tin từ thẻ Chứng minh nhân dân.
2. Tìm hiểu model nhận dạng vật thể sử dụng Single Shot Detector kết hợp mạng MobileNet, ResNet và áp dụng vào quá trình nhận dạng các thông tin trên thẻ.
3. Tìm hiểu model nhận diện chữ viết sử dụng mạng LSTM, Sequence to Sequence và áp dụng vào quá trình nhận dạng chữ.

Kế Hoạch Thực Hiện:

Thời gian	Kế hoạch	Nhiệm vụ
15/01 – 15/02	Học python	- Tìm hiểu cú pháp python. - Lập trình cơ bản python. - Sử dụng opencv trong python.

17/02 – 22/02	Học Machine Learning	- Học cấu trúc CNN. - Sử dụng thư viện keras xây dựng model CNN trong python.
24/02 – 29/02	Chuẩn bị dữ liệu	- Thu thập dữ liệu trên các trang mạng. - Tạo dữ liệu giả.
02/03 – 07/03	Xây model	- Tìm các source code model. - Kiểm tra tính chính xác.
09/03 – 14/03	Chạy model	- Thay đổi các trọng số, dữ liệu model phù hợp với tiếng Việt. - Chạy model và kiểm thử.
16/03 – 28/03	Hybrid methods	
30/03 – 04/04	Làm sạch dữ liệu	
06/04 – 11/04	Language modeling	- Xây dựng model học và chỉnh sửa các thông số có thể nhận diện sai.
13/04 – 15/07	Viết luận văn	

Xác nhận của GVHD	**Ngày…...tháng…...năm…...** **SV Thực hiện**

✪✪✪

LỜI CẢM ƠN

Đầu tiên, chúng em xin gửi lời cảm ơn chân thành và sâu sắc nhất đến thầy Lê Mai Tùng và thầy Lê Ngọc Thành vì đã tận tình hướng dẫn, kiên nhẫn và tạo động lực cho chúng em trong khoảng thời gian thực hiện luận văn này.

Chúng em xin gửi lời cảm ơn chân thành đến quý thầy cô trong Khoa Công nghệ Thông tin, Trường Đại học Khoa học Tự nhiên, cũng như tất cả quý thầy cô, anh chị hiện đang công tác bên ngoài đã tận tình giảng dạy chúng em trong suốt 4 năm qua. Đó không chỉ là kiến thức, mà còn là động lực giúp chúng em làm việc, nghiên cứu và vững bước trong tương lai.

Chúng em xin gửi lời cảm ơn chân thành đến quý Khoa và Nhà trường đã tạo điều kiện thuận lợi cho chúng em học tập và thực hiện luận văn này.

Chúng em xin gửi lời cảm ơn đến gia đình đã luôn quan tâm, động viên, ủng hộ về vật chất và tinh thần trong suốt thời gian qua.

Chúng em xin gửi lời cảm ơn đến tất cả bạn học trong và ngoài lớp đã đồng hành, động viên và giúp đỡ rất nhiều trong suốt thời gian học tập và làm việc.

Do thời gian và kiến thức có hạn nên không thể tránh khỏi những thiếu sót và hạn chế nhất định. Chúng em rất mong nhận được sự góp ý và chỉ dẫn của quý thầy cô và các bạn.

Cuối cùng, chúng em xin trân trọng cảm ơn và gửi lời chúc sức khỏe đến thầy cô, gia đình và bạn bè.

Thành phố Hồ Chí Minh, ngày tháng năm 2020

Sinh viên Sinh viên

MỤC LỤC

DANH MỤC HÌNH ẢNH

DANH MỤC BẢNG BIỂU

DANH MỤC THUẬT NGỮ

Thuật ngữ	**Chú giải**
Neural Networks	Mô hình mạng thần kinh
Artificial Neural Networks	Mô hình mạng thần kinh nhân tạo
Convolutional Neural Networks	Mô hình mạng thần kinh tích chập
Convolution	Phép tích chập
Pooling layer	Lớp tổng hợp
Bounding box	Hộp hình chữ nhật
Fully connected layer	Một lớp các khối mà tất cả đầu ra của các khối ở lớp trước đều là đầu vào của lớp sau
Over-fitting	Mô hình huấn luyện quá tốt quá phù hợp với dữ liệu huấn luyện, đến nỗi khi áp dụng với các dữ liệu kiểm tra đều không đưa ra kết quả tốt
Single Shot Multibox Detector	Mô hình phát hiện nhiều đối tượng thông qua các hộp biên trong một lần xử lý duy nhất.
Multibox	Nhiều hộp
Deep Neural Networks	Mô hình mạng thần kinh chuyên sâu
Depthwise Convolutions	Phép tích chập theo chiều sâu
Pointwise Convolutions	Phép tích chập theo điểm
Batch Normaliztion	Phương pháp chuẩn hóa các đặc điểm và phương sai của bộ dữ liệu
ReLU	Rectified Liner Unit, hàm lọc các giá trị bé hơn 0
Width multiplier	Tham số điều chỉnh số lượng kênh

Resolution multiplier	Tham số điều chỉnh mật độ đặc trưng điểm ảnh
Bottleneck	Lớp huấn luyện có ít hơn các khối mạng thần kinh dùng để thất thoát một vài đặc điểm ảnh, tránh tình trạng over-fitting
Maximun A Posteriori	Phương pháp đánh giá và tham đổi tham số dựa trên công thức xác xuất Bayes
Anchor	Các khung với kích thước khác nhau bao quanh một tập điểm đang được xem xét có là đối tượng hay không
GRU	Gated Recurrent Unit, là một phiên bản cải thiện của mạng RNN truyền thống
CMND	Chứng minh nhân dân, là loại giấy tờ tùy thân của công dân Việt Nam

TÓM TẮT KHÓA LUẬN

Với sự phát triển của các phương pháp học sâu nói riêng cũng như ngành thị giác máy tính nói chung đã tạo động lực giải quyết các bài toán được phát sinh trong thời đại số hóa, đặc trưng trong số đó là bài toán nhận diện ký tự quang học. Nội dung của luận văn này trình bày về nghiên cứu và xây dựng chương trình nhận diện và trích xuất thông tin từ thẻ Chứng minh nhân dân của công dân Việt Nam. Phương pháp tiếp cận được đề xuất là sử dụng các mô hình học máy để nhận diện, phân tách và trích xuất ký tự. Việc sử dụng các mô hình học máy sẽ cho kết quả nhận diện tốt hơn ở bất kể điều kiện thực tế khác nhau khi có thể được huấn luyện với tập dữ liệu đủ đa dạng. Thực nghiệm và đánh giá các mô hình cho thấy kết quả khả quan tạo tiền đề cho các phát triển tiếp theo trong tương lai.

CHƯƠNG 1: MỞ ĐẦU

1.1. Giới thiệu đề tài

Chứng minh nhân dân (hay chứng minh thư) là giấy tờ phân biệt quan trọng của mỗi người dân. Các thủ tục, công việc hay giao dịch liên quan đến chứng minh nhân dân là không thể tránh khỏi, đặc biệt là việc lấy và xác thực thông tin trên chứng minh nhân dân. Một vài việc thường thấy như chứng thực giấy tờ, đăng ký tài khoản, giao dịch với ngân hàng, đăng ký thẻ ngân hàng/ tín dụng, ... đến các công việc mức độ thấp hơn như trao đổi, buôn bán.

Trên thực tế, các công việc trên đều được thực hiện thủ công, trực tiếp từ nhân viên hoặc thông qua chủ sở hữu chứng minh thư, nên cần một khoảng thời gian cho mỗi lần thực hiện, và cần nhiều nguồn lực để có thể đẩy nhanh tiến độ, số lượng công việc.

Hiện nay, khi mọi người dần tiếp xúc, thực hiện các giao dịch nhiều hơn, các thông tin ngày càng lớn, các công việc liên quan đến chứng minh ngày càng nhiều. Nhưng vì nguồn nhân lực có hạn, thời gian hoạt động giới hạn, nên không thể thực hiện nhiều giao dịch, khiến các công việc liên quan bị chậm trễ, xảy ra nhiều hậu quả không nên có.

Dựa vào sự phát triển ngày càng vượt bậc của công nghệ, chúng ta nên "nhờ cậy" vào các thiết bị đơn giản xung quanh, cùng với khả năng tính toán của máy tính để hỗ trợ được nhiều hơn trong các công việc cần nhiều nguồn lực nhưng việc rút trích thông tin trên thẻ chứng minh nhân dân này.

1.2. Lý do chọn đề tài

Với nhu cầu đời sống hiện tại, việc sở hữu một thiết bị có khả năng chụp hình không khó. Dựa trên đó, nhóm chúng em quyết định nghiên cứu một giải pháp giúp thực hiện việc lấy thông tin trên chứng minh nhân dân thông qua những tấm hình bình thường, chất lượng không cao. Sau đó, có thể phân loại được các mục thông tin có trong thông tin như: số chứng minh nhân dân, họ tên, ngày sinh, ...

1.3. Nghiên cứu liên quan

Đọc và trích xuất văn bản trên hình ảnh không còn mới, đã có rất nhiều nghiên cứu và ứng dụng vào đời sống. Tuy nhiên, áp dụng các công nghệ đó cụ thể vào trích xuất thông tin trên chứng minh nhân dân không cho ra kết quả khả quan. Một trong những thư viện nhận diện ký tự quang học phổ biến hiện tại là Tesseract do Google phát triển. Nhưng phần đọc ký tự có dấu, cụ thể là tiếng Việt và phông chữ trên chứng minh nhân dân không được hỗ trợ nhiều, người dùng phải tự huấn luyện lại mô hình. Và để đạt được độ chính xác cao nhất, đầu vào cần là hình ảnh có độ tương phản cao giữa chữ và phong nền (thường là nền trắng chữ đen) và được tiền xử lý như làm rõ cạnh, làm mờ các điểm nhiễu ...

Dựa trên những lý do đó, trong đề tài này, luận văn vẫn giữ nguyên hướng tiếp cận, nhưng huấn luyện lại các mô hình máy học có sẵn để phù hợp hơn với chứng minh nhân dân. Hình ảnh đầu vào vẫn được tiền xử lý, xác định những vùng thông tin cần đọc. Sau đó, sử dụng các mô hình đã được huấn luyện phù hợp để đọc các ký tự trong những vùng thông tin đó.

1.4. Hướng tiếp cận và giải quyết vấn đề

Giải pháp đọc thông tin trên chứng minh nhân dân thông qua hình chụp được chia thành ba phần:

Phần thứ nhất gồm các công đoạn nhận dạng được vị trí chứng minh nhân dân có trong hình chụp. Hình chụp được tiền xử lý và sau đó cắt giữ mỗi chứng minh nhân dân.

Phần thứ hai, định vị các vùng chứa thông tin cần thiết và phân loại theo các mục có trong chứng minh nhân dân đã được đề trước.

Phần cuối cùng là công đoạn nhận diện các ký tự trong vùng chứa thông tin và chỉnh sửa các ký tự nhận dạng lỗi ở mức độ nhất định.

1.5. Mục tiêu luận văn

Các mục tiêu sinh viên tự đề ra để hoàn thành tốt đề tài luận văn:

- Xây dựng model nhận dạng chứng minh nhân dân trong hình chụp. Hình chụp không yêu cầu cao về chất lượng, nhưng phải bao quát được chứng minh nhân dân. Hình ảnh được tiền xử lý nếu cần.
- Khoanh vùng các phần loại thông tin trên hình chứng minh nhân dân đã cắt.
- Đọc các thông tin đã rút trích với model máy học được huấn luyện từ trước.
- Đạt độ chính xác cao nhất với số chứng minh nhân dân.

Một vài tiêu chí phụ hỗ trợ mức độ hoàn thiện hơn cho đề tài:

- Thời gian xử lý các phần trên trong khoảng cho phép (nhanh đến tương đối nhanh).
- Hỗ trợ tối đa ngôn ngữ tiếng Việt.
- Kích thước mô hình không quá lớn.

1.6. Phạm vi đề tài

Nhóm đã đặt mục tiêu và xác định rõ phạm vi đề tài chỉ bao gồm đọc các nội dung trong ảnh chụp chứng minh nhân dân. Bắt đầu từ nhận hình chụp bao quát được chứng minh nhân dân với chất lượng bình thường, và trả về kết quả là nội dung đã đọc được phân loại theo các mục. Nội dung chính được sử dụng là chữ số và ngôn ngữ tiếng Việt. Phạm vi không bao gồm đọc các loại thẻ khác hoặc đọc văn bản. Và không có ứng dụng cụ thể mà sẽ tích hợp thông qua API nếu phát triển trong tương lai.

1.7. Cấu trúc của luận văn

Chương 1 – Mở đầu: Nêu lên các vấn đề thường gặp và mức độ phổ biến của việc sử dụng chứng minh nhân dân của xã hội, đồng thời nêu lên những rắc rối và khó khăn kèm theo đó. Từ đó đưa ra vấn đề cần để giải quyết một phần hoặc toàn phần các khó khăn đó và phương pháp là đọc được các thông tin trên chứng minh nhân dân từ ảnh chụp có chủ đích. Chương này cũng nêu lên phạm vi và mục tiêu của luận văn.

Chương 2 – Cơ sở lý thuyết: Trình bày nội dung lý thuyết, cơ sở các Neural Network (mạng thần kinh nhân tạo) và các biến thể, các mô hình xác định vật thể, dự đoán, ... được ứng dụng vào đề tài.

Chương 3 – Nền tảng công nghệ: Giới thiệu môi trường huấn luyện và phát triển đề tài, các thư viện hỗ trợ và cách tạo các tập dữ liệu.

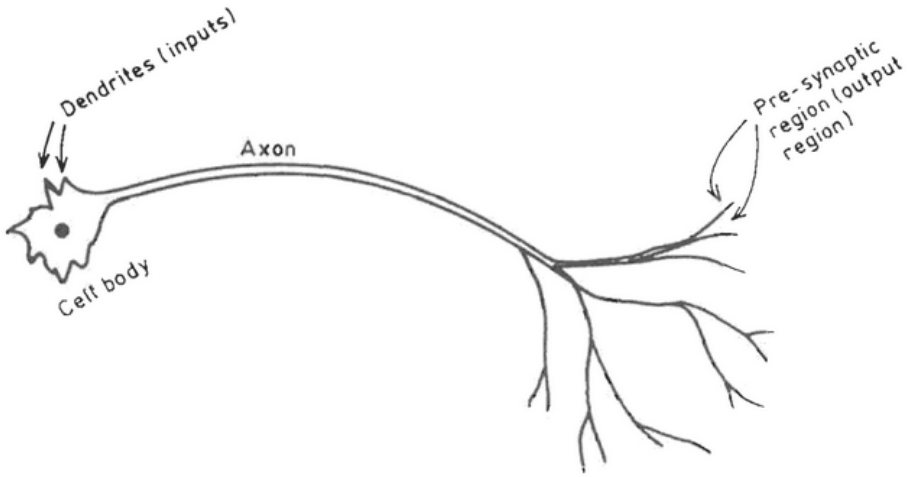
Chương 4 – Xây dựng giải pháp: Nêu lên giải pháp tiếp cận vấn đề được đặt ra. Trình bày chi tiết quy trình bao gồm: nhận diện thẻ chứng minh nhân dân trong ảnh, xác định những vùng dữ liệu quan trọng, và nhận diện trích xuất các ký tự từ những vùng đó.

Chương 5 – Thực nghiệm và đánh giá: Đưa ra kết quả, đánh giá so với thực tế của từng giai đoạn của đề tài.

Chương 6 – Kết luận: Trình bày các kết quả đạt được, các hạn chế, và ý tưởng cải thiện mô hình trong quá trình phát triển về sau.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1. Mô hình mạng thần kinh nhân tạo (Artificial Neural Networks – ANN)

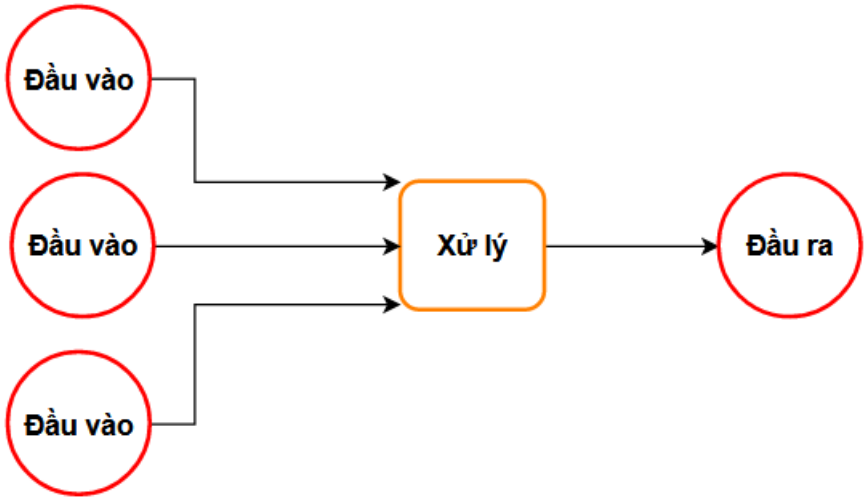


Hình 2.1 Tế bào thần kinh
(Nguồn: Daniel Graupe, Principles of Artificial Neural Networks, trang 5, Fig. 2.1. A biological neural cell (neuron).World Scientific 2013.)

Mạng thần kinh sinh học gồm các tế bào thần kinh (neural cell) kết nối với nhau. Một tế bào thần kinh bao gồm: nhánh sợi trục (dendrites), thân tế bào (cell body) và khớp tế bào (synaptic region). Trong đó, đa số các thao tác xử lý được vận hành trong thân tế bào. Nhánh sợi trục là nơi đưa thông tin, dữ liệu vào thân tế bào. Và khớp tế bào truyền kết quả đã được xử lý từ việc tổng hợp thông tin trong thân tế bào. Các thông tin đưa ra lại được xem như thông tin dữ liệu đầu vào truyền đến một tế bào thần kinh khác [1]. Sự lặp lại này hình thành một mạng thần kinh sinh học.

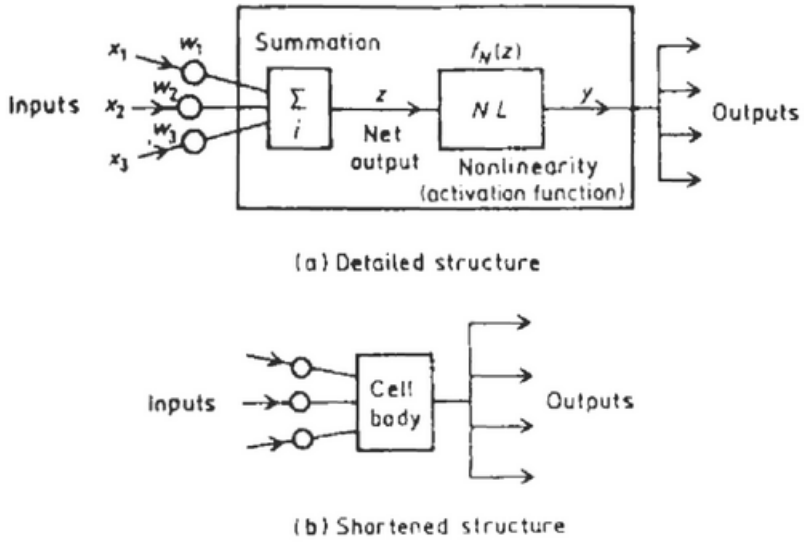
Là một thuật ngữ thường gặp trong lĩnh vực trí tuệ nhân tạo (Artificial Intelligence – AI) nói chung và cụ thể là máy học (Machine Learning – ML), mô hình mạng thần kinh nhân tạo (Artificial Neural Networks – ANN), hay còn gọi là mô hình mạng thần kinh (Neural Networks – NN) là một mô hình giúp hỗ trợ rất nhiều trong việc xử lý các bài toán có tính tuần tự, chuỗi và có tính chất sử dụng kết quả của phép tính trước vào phép tính sau. Mô hình mạng thần kinh và các biến thể

được ứng dụng rất nhiều vào các bài toán khó như phân tích nhận dạng hình ảnh, giọng nói hoặc xử lý ngôn ngữ tự nhiên, ...



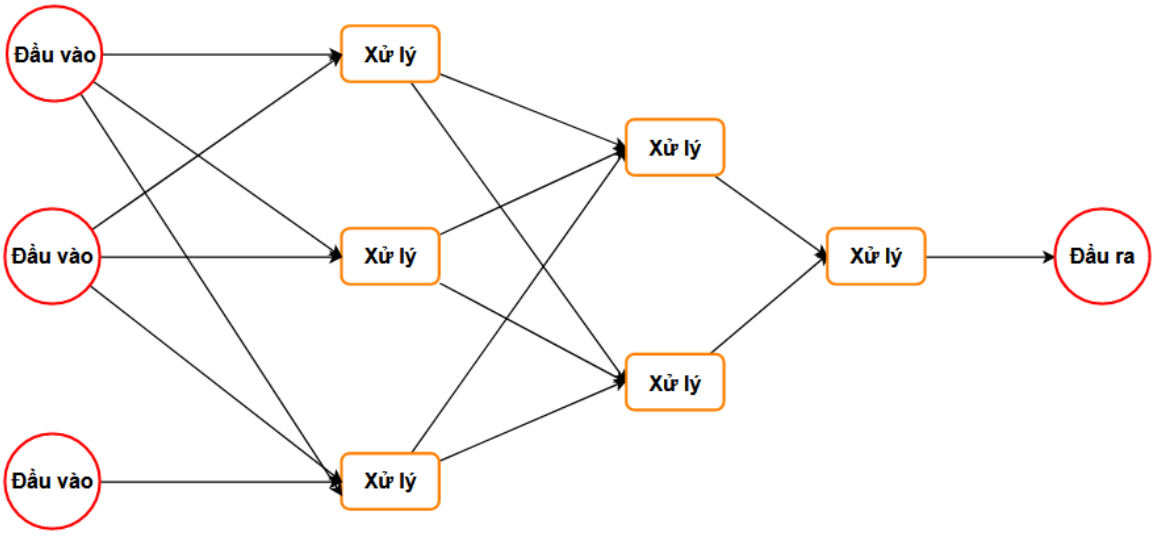
Hình 2.2 Một khối của mạng thần kinh nhân tạo

Mô hình mạng thần kinh được xây dựng dựa trên mạng thần kinh của con người, với nhiều đầu vào, được xử lý và đưa ra kết quả phù hợp nhất. 'Đầu vào' của một mô hình mạng thần kinh không giới hạn kiểu dữ liệu, cũng như không giới hạn số lượng, nhưng phải thống nhất cùng một loại. Trước khi được đưa vào xử lý, các dữ liệu được tiền xử lý như thêm trọng số,... để làm tăng hoặc giảm mức độ ảnh hưởng của dữ liệu. Đây là giai đoạn các dữ liệu vẫn độc lập tự thay đổi giá trị, khác với lúc đưa vào phần 'xử lý'. Dữ liệu sau khi đến phần xử lý sẽ được kết hợp với nhau, hoặc loại trừ,... thông qua phép tính trong phần xử lý, và tạo ra một dữ liệu mới với giá trị dựa vào các dữ liệu đầu vào. Đây là kết quả ở đầu ra.



Hình 2.3 Cấu trúc hoàn chỉnh của một khối
(Nguồn: Daniel Graupe, Principles of Artificial Neural Networks, trang 12, Fig. 3.2. A perceptron's schematic input/output structure. World Scientific 2013)

Cấu trúc hoàn chỉnh của một khối được biểu diễn trong Hình 2.3. Các dữ liệu trước khi được đưa vào xử lý được tiền xử lý thêm trọng số làm tăng hoặc giảm mức độ ảnh hưởng của thông tin đến việc xử lý. Sau đó, dữ liệu được tổng hợp lại và xử lý tùy theo hàm phi tuyến tính. Sau cùng, kết quả được trả về.



Hình 2.4 Mô hình mạng thần kinh nhân tạo

Hình 2.2 thể hiện một khối của mạng thần kinh đơn giản nhất. Và khi kết hợp nhiều khối này lại với nhau, với nguyên tắc đầu ra của khối trước được truyền đến –

một vài hoặc tất cả - các khối sau, ta được một mạng thần kinh nhân tạo, được biểu diễn đơn giản nhất như Hình 2.4.

Hình 2.4 biểu diễn một mạng thần kinh cơ bản đơn giản. Các kết quả quả việc 'xử lý' trước được đưa thành đầu vào của việc 'xử lý' sau. Số lượng 'xử lý' cũng không bị giới hạn, và không ràng buộc phần sau phải ít hoặc nhiều hơn phần trước. Riêng phần 'xử lý' cuối cùng chỉ có một và trả ra kết quả của toàn quá trình.

Dựa vào các đặc điểm của một khối và sự kết nối của chúng trong mô hình mạng thần kinh, công thức nguyên khai đầu tiên được đề xuất bởi nhà Tâm lý học Frank Rosenblatt vào năm 1958 (tương tự như Hình 2.3 a. Detailed structured) [2].

$$z = \sum_i \omega_i x_i$$

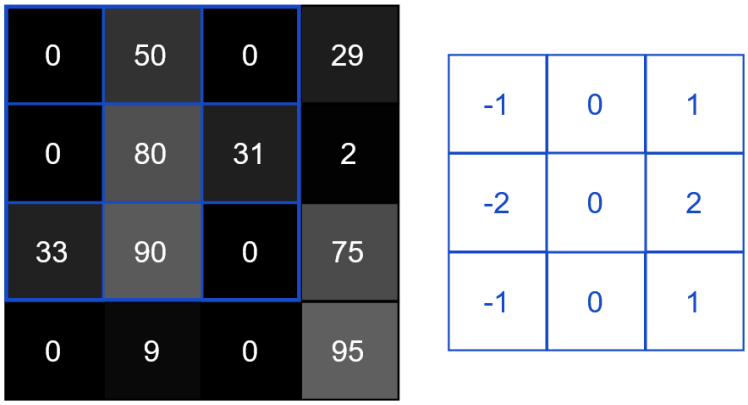
$$y = f_N(z)$$

Với x_i là các dữ liệu đầu vào, ω_i là trọng số của mỗi dữ liệu đó. z là tổng các dữ liệu đầu vào với từng trọng số. Hàm f_N là một hàm phi tuyến tính dùng để phân loại dữ liệu. Và y là dữ liệu trả về sau toàn bộ quá trình tính toán, được truyền đến các node sau (nếu có).

Trong quá trình phát triển trí tuệ nhân tạo, các cách kết nối giữa các khối, các công thức tính toán được thay đổi đa dạng (vẫn dựa trên công thức chính sơ khai), nhằm mục đích phù hợp hơn với đa số hay từng loại dữ liệu cụ thể.

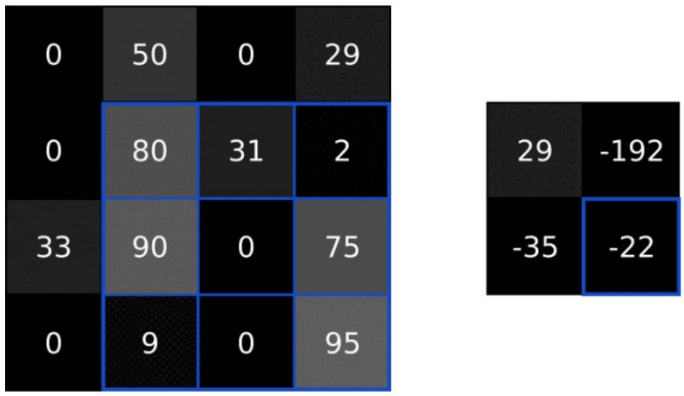
2.2. Mô hình mạng thần kinh tích chập (Convolutional Neural Networks – CNN)

Ứng dụng của mô hình mạng thần kinh nhân tạo ngày một đa dạng, và trở nên càng khó khăn khi phải tiếp xúc với dữ liệu kích thước rất lớn và không nhất quán như hình ảnh. Dựa trên cơ sở đó, mô hình mạng thần kinh tích chập (Convolutional Neural Networks - CNN) hình thành với mục đích giảm kích thước ảnh nhưng vẫn giữ được những nét đặc trưng thông qua phép tích chập (convolution).



Hình 2.5 Convolution filter (bên phải)
(Nguồn: https://victorzhou.com/blog/intro-to-cnns-part-1)

Mô hình mạng thần kinh gồm các lớp (tập hợp các khối) kết nối với nhau theo nguyên tắc đầu ra của lớp trước là đầu vào của lớp sau. Mô hình mạng thần kinh tích chập thì thêm một hoặc nhiều lớp tích chập và một hoặc nhiều lớp tổng hợp (pooling layer) [3].

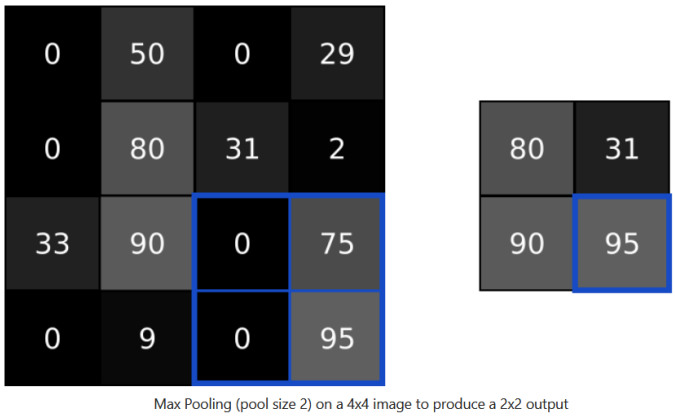


Hình 2.6 Kết quả thực hiện phép tích chập
(Nguồn: https://victorzhou.com/blog/intro-to-cnns-part-1)

Phép tích chập là một phép toán rất thường gặp trong xử lý ảnh. Kết quả của phép toán trả về làm rõ những đặc trưng nổi bật, và làm mờ hay thậm chí biến mất những điểm ảnh tầm thường, không nổi trội. Thường được dùng để xác định cạnh viền của đối tượng chính trong ảnh. Phép tích chập bao gồm một tấm nền (filter) chứa những trọng số tùy chỉnh theo kích thước tùy chỉnh.

Một lớp tích chập bao gồm một hoặc nhiều tấm nền tích chập. Như vậy, với một tấm hình, ta sẽ được n – hình sau khi được tích chập phụ thuộc vào số lượng tấm nền tích chập được thêm vào.

Sau khi thực hiện tích chập, tuy rằng kích thước dữ liệu còn lớn hơn, nhưng dữ liệu đã được chuyển về giữ những đặc điểm nổi trội của tấm hình tùy theo tấm nền tích chập. Và thường sau lớp tích chập, mô hình sẽ kèm theo lớp tổng hợp, dùng để lấy những đặc trưng nổi trội nhất của tấm hình [4].



Hình 2.7 Kết quả thực hiện phép tổng hợp (Max pooling)
(Nguồn: https://victorzhou.com/blog/intro-to-cnns-part-1)

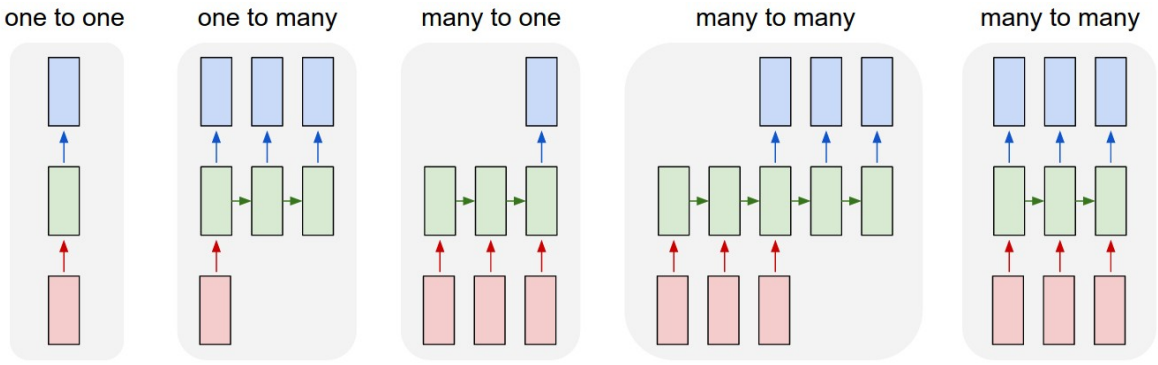
Như vậy, sau khi thực hiện phép tổng hợp, kích thước dữ liệu sẽ được giảm đi n-lần phụ thuộc vào lớp tổng hợp $n * n$, nhưng vẫn giữ được đặc trưng nổi trội nhất của một tấm hình.

Cuối cùng, mô hình thông thường sẽ dùng thêm một lớp fully connected để kết nối các đặc trương của một tấm hình lại, lưu trữ tính chất và trả về kết quả.

Việc sử dụng lớp tổng hợp sẽ làm giảm kích thước của dữ liệu, và sử dụng lớp tích chập để lấy những điểm nổi bật. Tuy nhiên, cần lưu ý nếu sử dụng nhiều lớp tổng hợp hoặc sử dụng quá nhiều lớp tích chập sẽ dẫn đến đầu vào chỉ còn là những đặc trưng tiêu biểu nhất, gây nên trường hợp over-fitting. Nhưng nếu sử dụng quá ít lớp tích chập thì sẽ dẫn đến mỗi tấm hình đều có nhiều đặc trưng, bao gồm cả những đặc trưng không cần thiết, và gây nhiễu cho kết quả đầu ra.

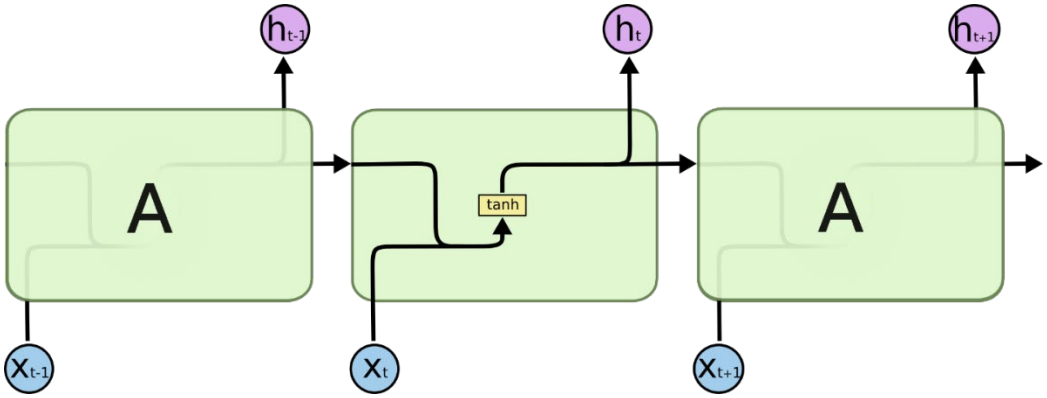
2.3. Mô hình mạng thần kinh hồi quy (Recurrent Neural Networks – RNN)

Xử lý thông tin dạng chuỗi là điểm mạnh của mô hình mạng thần kinh nhân tạo, và trình tự là một trong những đặc trưng nổi bật của chuỗi. Từ đó, mô hình mạng thần kinh hồi quy (Recurrent Neural Networks – RNN) ra đời nhằm khai thác đặc trưng này. Ngoài ra, với mạng thần kinh tích chập, dữ liệu đầu vào và kết quả trả về là cố định, nhưng với mạng thần kinh hồi quy thì dữ liệu đầu vào thay đổi được và kết quả trả về được phép nhiều hơn một.



Hình 2.8 Vài dạng về mô hình mô hình mạng thần kinh hồi quy
(Nguồn: https://karpathy.github.io/2015/05/21/rnn-effectiveness/)
với màu đỏ là đầu vào, màu xanh lá là một node của Recurrent Neural Networks, và màu xanh dương là đầu ra

Trong mô hình mạng thần kinh hồi quy, các khối ngoài nhận dữ liệu đầu vào, còn truyền dữ liệu đến các nhanh kề cận theo trình tự (cùng chiều hoặc ngược chiều trình tự chuỗi, hoặc cả hai – Bidirectional Recurrent Networks). Hình 2, một khối nhận vào gồm dữ liệu đầu ra được truyền từ khối trước đó và X_t – dữ liệu truyền vào. Sau khi được tổng hợp, dữ liệu đầu vào được chọn lọc thông qua hàm tanh hoặc một hàm sigmoid khác. Cuối cùng, dữ liệu trả ra gồm đầu ra h_t và được truyền đến khối tiếp theo. Như vậy mạng thần kinh hồi quy nhận vào được một chuỗi có trình tự và trả ra một chuỗi cũng có trình tự.

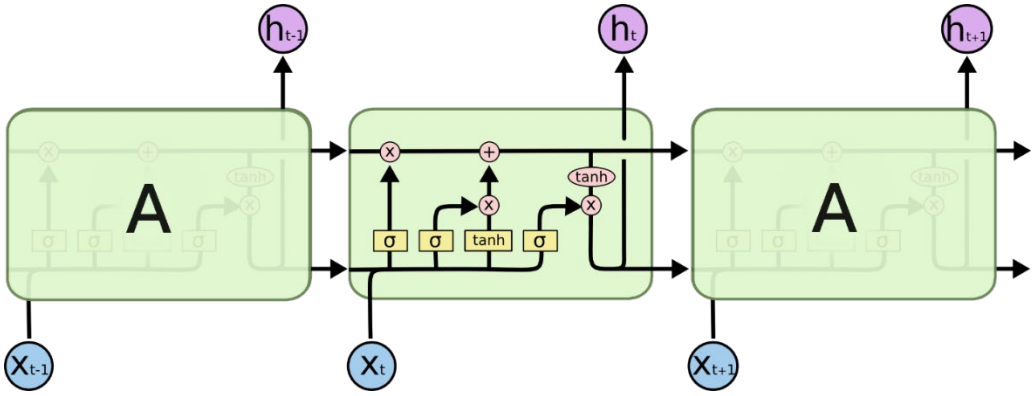


Hình 2.9 Cấu trúc một khối của mạng thần kinh hồi quy
(Nguồn: https://colah.github.io/posts/2015-08-Understanding-LSTMs/)

Tuy nhiên, việc truyền đạt thông tin từ các khối cận kề là hữu hạn. Vì nếu thông tin ở node A_{t-4} có ảnh hưởng đến thông tin ở node A_t,, nhưng lại không ảnh hưởng đến thông tin ở các node A_{t-5}, A_{t-3}, ... xung quanh, thì cũng sẽ bị giảm độ quan trọng dần thậm chí bị loại bỏ. Trong bài viết [5] của mình, Yoshua Bengio, Patrice Simard, và Paolo Frasconi đã chứng minh được điều này.

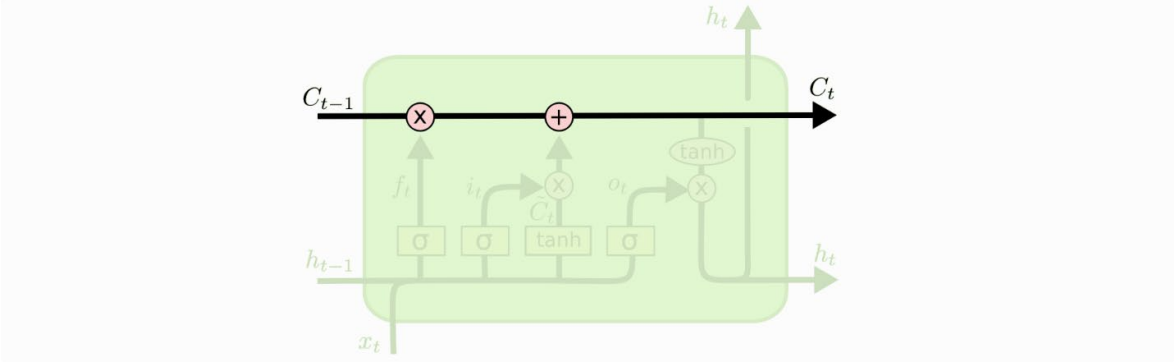
2.4. Mô hình bộ nhớ dài hạn – ngắn hạn (Long Short Term Memory - LSTM)

Mô hình mạng thần kinh hồi quy hỗ trợ xử lý đến các phụ thuộc xung quanh một khối, nhưng các phụ thuộc đó nếu quá xa sẽ không còn nhiều tác động đến khối đó nữa. Việc thay đổi trọng số tăng giảm mức độ ảnh hưởng của dữ liệu phần nào sẽ giúp được việc này. Tuy nhiên, Mô hình bộ nhớ dài hạn – ngắn hạn (Long Short Term Memory – LSTM) lần đầu được giới thiệu bởi Sepp Hochreiter và Jurgen Schmidhuber – đã giải quyết được vấn đề này [6]. Là một trong những biến thể quan trọng của Mô hình mạng hồi quy, mô hình bộ nhớ dài hạn – ngắn hạn đến nay vẫn được phát triển và gần nhất vào năm 2019.



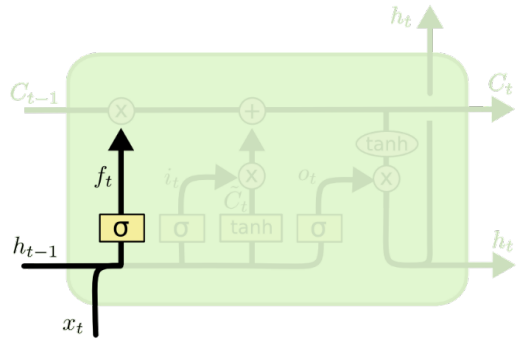
Hình 2.10 Kiến trúc một khối trong mô hình bộ nhớ dài hạn – ngắn hạn
(Nguồn: https://colah.github.io/posts/2015-08-Understanding-LSTMs/)

Để truyền được các dữ liệu từ xa, và cũng là đặc điểm nổi bật làm nên mạng mô hình bộ nhớ dài hạn – ngắn hạn, đường trạng thái tế bào (cell state). Đường trạng thái tế bào chạy xuyên suốt trong mạng, nên thông tin cũng sẽ được truyền và được cập nhật bởi tất cả các khối. Tuy nhiên, mô hình bộ nhớ dài hạn – ngắn hạn sẽ sàng lọc thông tin cẩn thận trước khi thêm hoặc xóa bớt bởi ba cổng, nên thông tin không thay đổi nhiều, bảo toàn mức nguyên vẹn của thông tin.



Hình 2.11 Đường trạng thái tế bào
(Nguồn: https://colah.github.io/posts/2015-08-Understanding-LSTMs/)

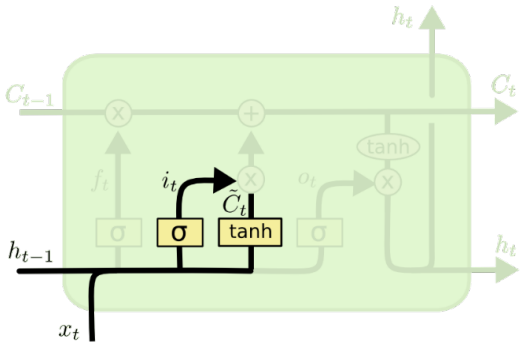
Các cổng tương tác với đường trạng thái tế bào như sau. Ở cổng đầu tiên, cổng quên (forgot gate), khối nhận thông tin dữ liệu từ khối trước (h_{t-1}) và thông tin nhập vào (x_t). Dựa vào tính toàn, cổng trả về lượng dữ liệu sẽ bị lược bỏ trên đường cell state. Quá trình này được gọi là Tầng cổng quên (Forgot gate layer).



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

Hình 2.12 Tầng cổng quên
(Nguồn: https://colah.github.io/posts/2015-08-Understanding-LSTMs/)

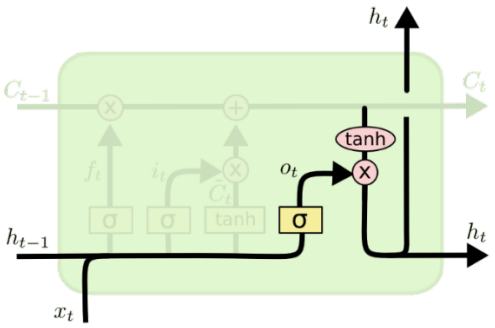
Tiếp theo, mạng bộ nhớ dài hạn – ngắn hạn dùng hàm sigmoid trong cổng cập nhật (update gate) và tanh để sàng lọc và tạo ra một vector giá trị dữ liệu mới dùng để cập nhật vào đường trạng thái tế bào.



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Hình 2.13 Tầng cổng cập nhật
(Nguồn: https://colah.github.io/posts/2015-08-Understanding-LSTMs/)

Cuối cùng, dữ liệu được sàng lọc một lần nữa thông qua cổng đầu ra (output gate), kết hợp với thông tin trong đường trạng thái tế bào, và được trả về cũng như truyền đến tế bào tiếp theo. Đây là tầng cổng đầu ra (output gate layer).



$$o_t = \sigma\left(W_o \; [h_{t-1}, x_t] \; + \; b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

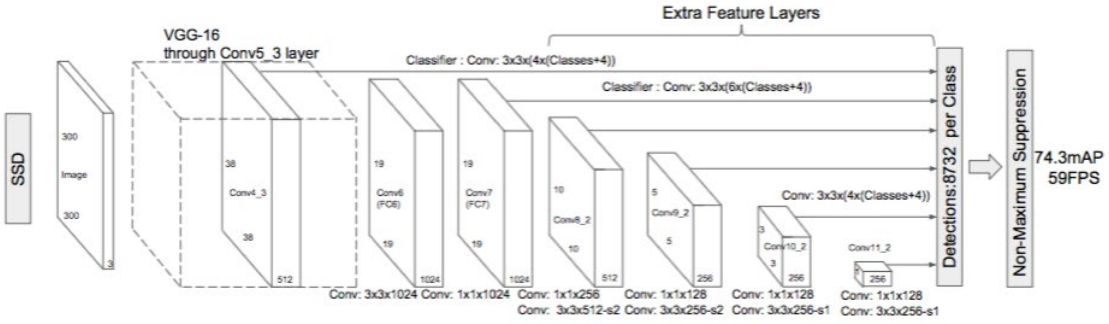
Hình 2.14 Tầng cổng đầu ra
(Nguồn: https://colah.github.io/posts/2015-08-Understanding-LSTMs/)

2.5. Single Shot Multibox Detector (SSD)

Một trong những bài toán phổ biến của trí tuệ nhân tạo là nhận diện vật thể. Và mô hình SSD với nhiều ưu điểm đã phần nào giúp việc giải quyết bài toán trên [7]. Với đặc điểm nhận biết và phân loại đối tượng thông qua một giai đoạn duy sử dụng kỹ thuật multibox. SSD trả về kết quả nhanh hơn rất nhiều nhờ thực hiện ít công đoạn hơn, và có thể đáp ứng được thời gian thực. Kết quả trả về của SSD so với các thuật toán khác (R-CNN, Fast R-CNN, Faster R-CNN) cho thời gian thực hiện và kết quả ấn tượng hơn.

Hình 2.15 Bounding box dựa vào texture
(Nguồn: van de Sande et al. ICCV'11)

Điểm nhấn của mô hình này là sử dụng các hộp biên (Bounding box) để xác định đối tượng trong hình [8]. Đặc trưng của hộp biên là các khung xác định đối tượng thay đổi lớn nhỏ phù hợp tùy loại đối tượng. Vì thế, SSD có xuất hiện từ "Multibox", ý chỉ sự đa dạng về kích thước khung xác định đối tượng. Với mỗi điểm ảnh, mô hình sẽ hình thành các hộp biên phù hợp bao xung quanh "đối tượng tạm thời", từ số điểm ảnh tăng dần, và số lượng hộp biên được thiết lập từ trước. "Đối tượng tạm thời" được hình thành bằng các nhóm các điểm ảnh, mà ngữ cảnh (texture) (kết cấu như màu sắc, độ tương phản,...) gần nhau.



Hình 2.16 Cấu trúc mô SSD
(Nguồn: SSD: Single Shot MultiBox Detector [7])

Mô hình SSD rút trích đặc trưng của ảnh thông qua VGG-16 [9]. Sau đó, mô hình lần áp dụng các lớp tích chập, tổng hợp và fully connected để giảm bớt các điểm ảnh nhưng vẫn giữ được đặc trưng của tấm hình.

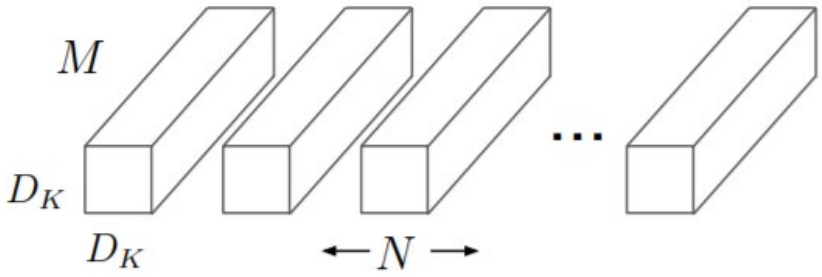
$$L(x, c, l, g) = \frac{1}{N}(L_{conf}(x, c) + \alpha L_{loc}(x, l, g))$$

Hình 2.17 Hàm mất mát của SSD
(Nguồn: https://towardsdatascience.com/review-ssd-single-shot-detector-object-detection-851a94607d11)

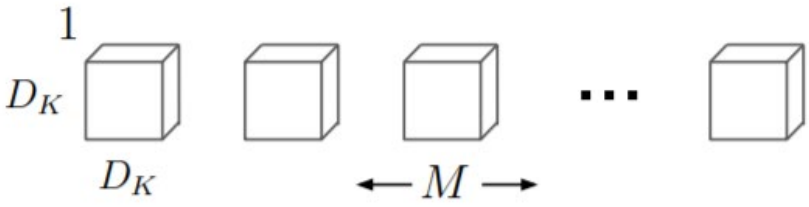
Trong quá trình học, mô hình sử dụng hàm mất mát (loss function) để cải thiện độ chính xác. Với $L_{con\,f}(x, c)$ là độ tin cậy hàm mất mác của x (hộp biên) và c (số lớp đối tượng), và $L_{loc}(x, l, g)$ là hàm lỗi cục bộ (Localization loss function) giữa l (khung dự đoán) và g (khung chính xác) của nó.

2.6. Mạng trích xuất đặc trưng MobileNet

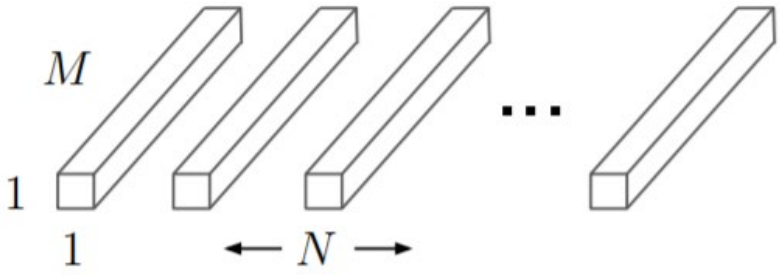
Với ý tưởng rút ngắn kích thước mô hình nhưng vẫn trả về kết quả tốt, nhóm các tác giả đến từ Google đã thiết kế cấu trúc mô hình MobileNet (hiện tại đã là phiên bản thứ hai) nhằm có thể chạy Deep Neural Networks trên nền tảng các thiết bị cấu hình yếu. Bằng cách áp dụng phép tích chập phân tách theo chiều sâu (Depthwise Separable Convolution) [10], cấu trúc mô hình đã giảm bớt nhiều tham số và độ phức tạp, nhưng độ chính xác vẫn không ảnh hưởng nhiều.



(a) Standard Convolution Filters



(b) Depthwise Convolutional Filters



(c) 1×1 Convolutional Filters called Pointwise Convolution in the context of Depthwise Separable Convolution

Hình 2.18 Cấu trúc lớp tích chập thông thường (a) và lớp tích chập phân tách theo chiều sâu(b và c)

(Nguồn: MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications)

Giả sử đầu vào có M kênh (channels), và đầu ra có N kênh, với kích thước một kernel là $D_k * D_k$, và kích thước một feature map $D_F * D_F$, ta có tổng số phép tính phải thực hiện trong cấu trúc tích chập thông thường:

$$D_K * D_K * M * N * D_F * D_F$$

Lớp tích chập phân tách theo chiều sâu được tách thành hai lớp: Depthwise Convolutions và Pointwise Convolutions. Depthwise Convolutions dùng để áp dụng từng filter $(D_k * D_k)$ vào M kênh của đầu vào:

$$D_K * D_K * M * D_F * D_F$$

Và Pointwise Convolutions đơn giản tạo ra sự kết hợp tuyến tính từ kết quả trả về của lớp Depthwise Convolutions thành N kênh đầu ra:

$$D_K * D_K * \ N * D_F * D_F$$

Kết hợp Depthwise Convolutions và Pointwise Convolutions thành Depthwise Separable Convolution:

$$D_K * D_K * M * D_F * D_F + \ D_K * D_K * \ N * D_F * D_F$$

Khi so sánh với lớp tích chập thông thường, Depthwise Separable Convolution thể hiện tốc độ nhanh hơn nhiều $(\frac{1}{N} + \frac{1}{D_K^2})$.

MobileNet được xây dựng dựa trên Depthwise Separable Convolution, cùng với cấu trúc như hình 19, sử dụng Batch Normalization và ReLU sau mỗi lớp tích chập. Ngoài ra, MobileNet còn sử dụng tham số *alpha* (width multiplier) điều khiển số lượng channel và tham số *rho* (resolution multiplier) điều khiển mật độ đặc trưng điểm ảnh. Như vậy, mô hình sẽ được giảm được các phép tính, thông số và kích thước.

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$ Conv dw / s1 Conv / s1	$3 \times 3 \times 512$ dw $1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$ $14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Hình 2.19 Cấu trúc mô hình MobileNet
(Nguồn: MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications, bảng 2)

So với phiên bản tiền nhiệm, MobileNetV2 có thêm Basic Building block (Bottleneck depth-separable Convolution với Residuals) [11]. Trong đó, Basic Building block sử dụng lớp Bottleneck để giảm bớt số lượng kênh với hàm biến đổi phi tuyến tính thay vì dùng tham số alpha. Kết quả thực nghiệm với SSDLite được mô tả ở bảng 2.1, tuy MobileNetV2 có độ ước lượng Maximun A Posteriori thấp hơn (0,1%) nhưng lại có thời gian thực hiện nhanh hơn.

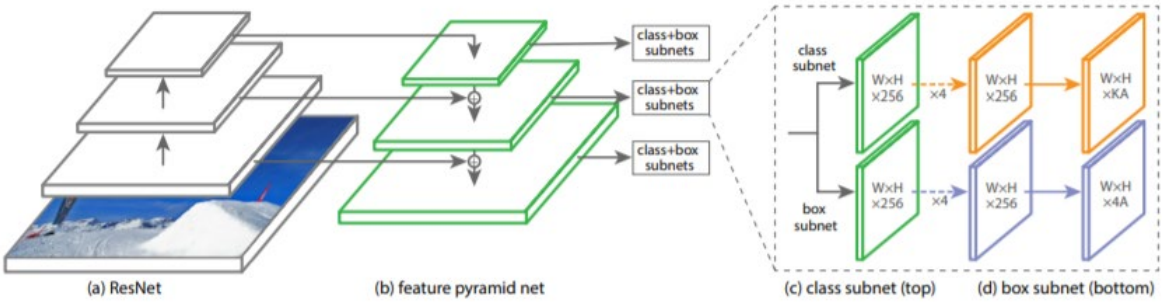
Bảng 2.1 Bảng so sánh MobileNetV1 và MobileNetV2

Model	Params	Multiply-Adds	mAP	Mobile CPU
MobileNetV1 + SSDLite	5.1M	1.3B	22.2%	270ms
MobileNetV2 + SSDLite	4.3M	0.8B	22.1%	200ms

(Nguồn: https://ai.googleblog.com/2018/04/mobilenetv2-next-generation-of-on.html)

2.7. Mạng trích xuất đặc trưng RetinaNet

RetinaNet là một mô hình mạng duy nhất, bao gồm một mạng chính (backbone network) và hai mạng phụ (subnetworks) với nhiệm vụ cụ thể. Trong đó, mạng chính là mạng tích chập tự động (off-the-self convolution network) áp dụng mạng đặc trưng xoáy (conv feature map) cho toàn ảnh đầu vào. Mạng phụ thứ nhất phân loại dựa trên kết quả của mạng chính. Mạng phụ thứ hai thực hiện hồi quy tích chập khung giới hạn (convolution bouding box regression) [12].



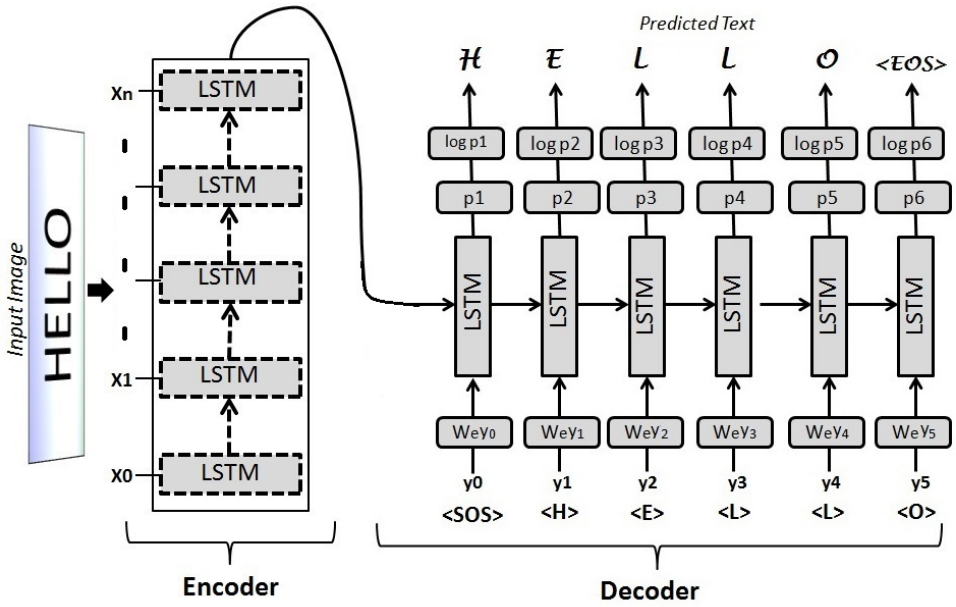
Hình 2.20 Cấu trúc RetinaNet
(Nguồn: Focal Loss for Dense Object Detection [12])

Nhóm tác giả sử dụng mạng Feature Pyramid Network làm mạng chính [13]. Mạng chính sẽ rút trích đặc trưng và trả về tập dữ liệu gồm các vùng đặc trưng và anchor của từng vùng đó. Mạng phụ thứ nhất sẽ dự đoán xác suất mà vùng đó có là một đối tượng hay không cho từng anchor và từng lớp đối tượng của anchor đó. Các thông số đánh giá được chia sẻ với mạng chính để thay đổi phù hợp hơn trong quá trình huấn luyện. Mạng phụ thứ hai tồn tại song song với mạng phụ thứ nhất, với cấu trúc tương tự như mạng phụ thứ nhất nhưng lại hồi quy số liệu của mỗi anchor box đến đến đối tượng mẫu (ground-truth object) gần nó nhất, nếu có. Các tham số trong mạng phụ thứ hai không được chia sẻ đến mạng chính mà được dùng để tự điều chỉnh và đánh giá. Trong quá trình huấn luyện, mô hình áp dụng Focal Loss để tự đánh giá và thay đổi tham số [13].

2.8. Mô hình chuyển đổi chuỗi sang chuỗi (sequence to sequence – seq2seq)

Sequence to sequence (còn được gọi với tên encoder-decoder) [14] là mô hình được tạo ra để giải quyết các bài toán sinh chuỗi từ việc học các chuỗi đầu vào cho trước, do đó nó phục vụ hiệu quả cho việc dịch máy [15]. Ngày nay nó dần được áp dụng rộng rãi trong các hệ thống khác như nhận dạng giọng nói [16], tóm tắt văn bản [17], thêm mô tả cho ảnh [18] …

Mô hình gồm hai phần chính là phần mã hóa (Encoder) và phần giải mã (Decoder) đều được tạo thành từ các mạng như RNN, LSTM, GRU [15]. Cụ thể được áp dụng trong luận văn này là mạng LSTM và đầu vào của Encoder là các đặc trưng ảnh chứa từ khóa được trích xuất từ tầng CNN.



Hình 2.21 Mô hình Encoder Decoder cho nhận dạng kí tự quang học (Nguồn: https://www.groundai.com/project/sequence-to-sequence-learning-for-optical-character-recognition/2 [19])

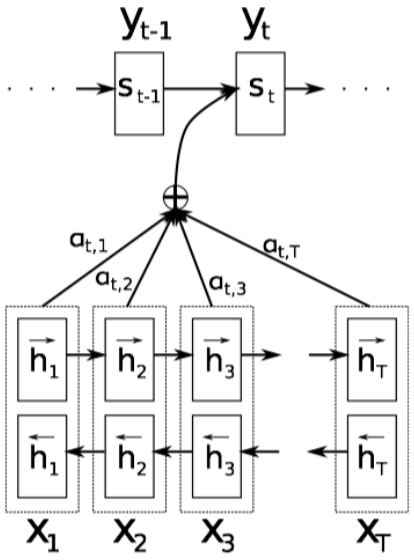
Phần Encoder mã hóa các đặc trưng ảnh đầu vào bằng một chuỗi các lớp LSTM và đưa ra vec-tơ đại diện cuối cùng, vec-tơ này cũng là đầu vào của phần Decoder. Ở đây mô hình LSTM được sử dụng vì nó giải quyết được vấn đề bộ nhớ ngắn hạn gặp phải ở mô hình RNN. Nhờ việc lưu trữ các thông tin được xử lý ở các ô trạng thái (cell state), LSTM có thể mang các thông tin của ảnh từ khóa đi xa.

Phần Decoder nhận vec-tơ đại diện cuối cùng của Encoder như một trạng thái khởi tạo ban đầu. Tiếp đến sẽ thực hiện tính toán các xác suất có điều kiện của thông tin đầu vào so với các kí tự trong bảng mã và chọn ra kí tự có xác suất cao nhất. Xác suất này được truyền qua các lớp LSTM và kết quả cuối cùng sẽ lấy tích xác suất của các kí tự đề chọn ra từ có tích xác suất cao nhất.

2.9. Kỹ thuật chú ý (Attention) trong mô hình chuyển đổi chuỗi sang chuỗi

Mô hình Seq2Seq mang lại sự hiệu quả cao trong các bài toán dịch máy nhưng nó lại có vấn đề tiềm ẩn khi dự đoán các chuỗi dài, đặc biệt là các chuỗi dài hơn so với các chuỗi trong tập huấn luyện. Nguyên nhân của vấn đề này là vì mô hình cần phải mã hóa toàn bộ thông tin đầu vào thành một vec-tơ duy nhất có độ dài cố định [15].

Kỹ thuật Attention được áp dụng trong mô hình để khắc phục nhược điểm này bằng việc giúp mô hình có thể tập trung vào những phần quan trọng của dữ liệu đầu vào thay vì chỉ sử dụng vec-tơ đại diện cuối cùng của phần Encoder [15].



Hình 2.22 Cơ chế phát sinh từ y_t dựa trên chuỗi đầu vào
(Nguồn: Neural Machine Translation by Jointly Learning to Align and Translate [15])

Cơ chế này sử dụng tất cả các kết quả đầu ra của từng cell kết hợp với trạng thái ẩn tương ứng để đưa ra một vec-tơ ngữ cảnh (attention vec-tơ) làm đầu vào cho

từng cell trong Decoder. Công thức tính attention vec-tơ vào thời gian t cho mỗi từ đầu vào $(1, \dots, T_A)$ được biểu diễn như sau:

$$u_i^t = v^T \tanh(W_1' h_i + W_2' d_t)$$

$$a_i^t = softmax(u_i^t)$$

$$d_t' = \sum_{i=1}^{T_A} a_i^t h_i$$

Với $(h_1, \dots, h_{T_A})$ là trạng thái ẩn của Encoder, $(d_1, \dots, d_{T_B})$ là trạng thái ẩn của Decoder. Vec-tơ v và các ma trận W_1', W_2' là các tham số có thể học được từ mô hình. Vec-tơ u^t có độ dài bằng T_A và mỗi phần tử chứa trọng số mà cơ chế cần áp dụng lên mỗi phần tử tương ứng của trạng thái ẩn phần Encoder [7].

CHƯƠNG 3: NỀN TẢNG CÔNG NGHỆ

3.1. Môi trường phát triển

3.1.1 Các thư viện hỗ trợ

Để tối ưu hóa thời gian nghiên cứu và cài đặt các mô hình bằng ngôn ngữ lập trình Python, luận văn sử dụng một số thư viện hỗ trợ rất tốt cho các bài toán học máy và xử lý ảnh, điển hình là thư viện Tensorflow và OpenCV.

a. Thư viện Tensorflow:

Hình 3.1 Thư viện Tensorflow
(Nguồn: https://www.tensorflow.org)

Tensorflow là thư viện mã nguồn mở do đội ngũ Google Brain phát triển, hỗ trợ rất lớn trong việc nghiên cứu và ứng dụng trí tuệ nhân tạo, máy học,… Thư viện được phát triển bằng ngôn ngữ C++ nên hiệu năng được trau dồi rất tốt và được hỗ trợ để có thể chạy trên cả CPU hay GPU, trên một server lơn hay đơn giản làm một chiếc điện thoại thông minh nhỏ gọn hiện nay.

Thư viện này cung cấp đầy đủ các nền tảng hỗ trợ người dùng xây dựng, huấn luyện và triển khai các mô hình học máy một cách dễ dàng cho các bài toán như xử lý ảnh, xử lý giọng nói, văn bản…

Tensorflow được ứng dụng phổ biến trong các dịch vụ của Google như phân loại email, phân loại hình ảnh và văn bản, nhận biết khuôn mặt, tối ưu hóa kết quả tìm kiếm,… Không giới hạn lại ở đó, một số dự án nổi tiếng trên thế giới cũng dùng thư viện này.

Trong phạm vi phát triển và nghiên cứu của luận văn này, phiên bản Tensorflow được sử dụng là phiên bản 1.15 dùng để xây dựng, huấn luyên và kiểm thử các mô hình nhận diện đối tượng cũng như mô hình nhận diện ký tự.

b. Thư viện OpenCV:

OpenCV cũng là một thư viện mã nguồn mở hàng đầu về xử lý ảnh, thị giác máy tính, máy học,… Thư viện hỗ trợ giao tiếp với các ngôn ngữ phổ biến như C/C++, Java, Python và hầu hết các hệ điều hành hiện nay. Thư viện được thiết kế để tỉnh toán hiệu quả và tập trung vào các ứng dụng thời gian thực.

Hình 3.2 Thư viện OpenCV
(Nguồn: https://opencv.org)

Một số ứng dụng mà OpenCV được áp dụng rộng rãi như: Nhận diện hình ảnh, Kiểm tra và giám sát tự động, Robot và xe tự hành, Tìm kiếm và phục hồi hình ảnh/video, Phân tích hình ảnh y tế,…

3.1.2 Môi trường cài đặt và huấn luyện

Các mô hình trong luận văn được huấn luyện và thực nghiệm trên Google Colab (hay còn gọi là Google Colaboratory). Đây là một nền tảng miễn phí do Google Research phát triển nhằm giúp người dùng học tập, nghiên cứu về các chủ đề trí tuệ nhân tạo, học máy, khoa học dữ liệu,… thay vì phải chọn các giải pháp đắt đỏ khác như thuê dịch vụ điện toán trên AWS.

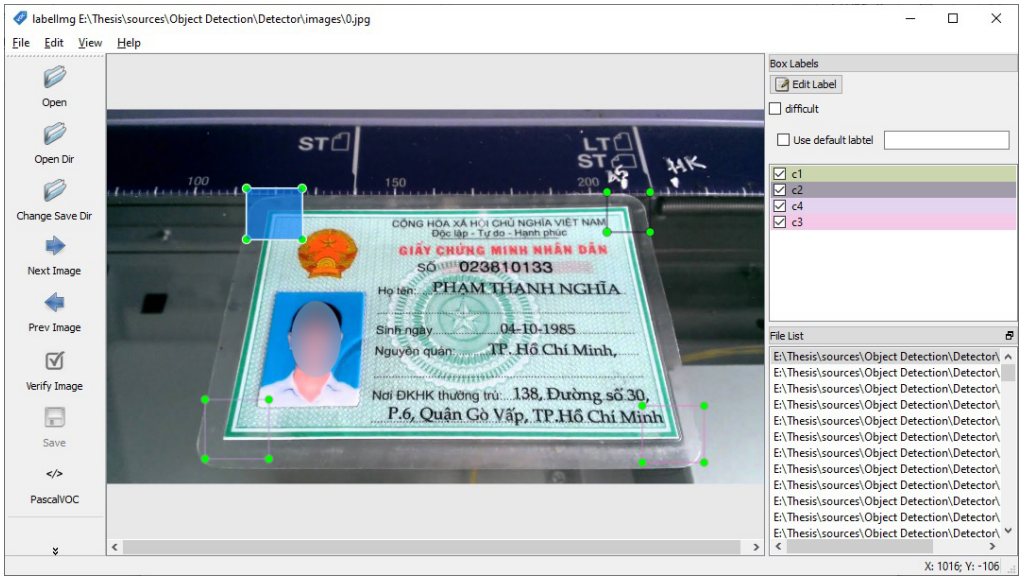
Hình 3.3 Logo nền tảng Google Colab
(Nguồn: https://colab.research.google.com)

Colaboratory cho phép chương trình Python được thực thi thông qua nền tảng đám mây với các tài nguyên máy tính như CPU, GPU, TPU tốc độ cao. Do được phát triển trên nền tảng Jupyter Notebook nên việc sử dụng rất đơn giản và trực quan. Bên cạnh đó Google Colab cũng được cài đặt sẵn nhiều thư viện phổ biến để nghiên cứu Deep Learning như Tensorflow, Keras, OpenCV,…

Cấu hình môi trường huấn luyện và kiểm thử các mô hình bao gồm CPU 2.30 GHZ với 13 GB Ram và GPU Tesla P4 với 12 GB VRam đủ để huấn luyện các mô hình có kích thước tập dữ liệu lớn.

3.1.3 Chương trình gán nhãn hình ảnh

Trong quá trình tạo dữ liệu huấn luyện cho các mô hình nhận diện đối tượng, LabelImg là một chương trình không thể thiếu đã giúp việc gán nhãn các đối tượng được thực hiện một cách nhanh chóng và hiệu quả. Đây là chương trình mã nguồn mở do cộng đồng lập trình viên duy trì và phát triển, hỗ trợ việc gán nhãn các đối tượng theo khung hình chữ nhật và lưu theo hai loại định dạng phổ biến là PascalVOC và YOLO.



Hình 3.4 Gán nhãn các đối tượng bằng chương trình LabelImg

Việc gán nhãn các đối tượng được minh họa như hình 3.3 với các thao tác đơn giản như giữ và kéo chuột từ điểm đầu đến điểm cuối và nhập nhãn cần gán. Các tập tin nhãn được lưu ở định dạng Text hoặc XML tùy thuộc vào định dạng người dùng chọn.

3.2. Tạo dữ liệu huấn luyện

Phần lớn ảnh CMND đóng vai trò là dữ liệu huấn luyện và kiểm thử được phát sinh từ chương trình tạo dữ liệu. Chương trình hoạt động bằng việc sử dụng các phôi thẻ CMND mặt trước và mặt sau trắng thông tin, sau đó phát sinh các trường thông tin liên quan như số CMND, họ tên, ngày sinh, địa chỉ, thời gian cấp, nơi cấp và ghi lên các vị trí cố định trên thẻ. Hiện nay có hai loại thẻ CMND là loại thẻ cũ – cấp trước năm 2010 và loại thẻ mới – cấp từ năm 2010 trở về sau.

Hình 3.5 Một mẫu phôi thẻ CMND

Trường số CMND là một chuỗi được phát sinh ngẫu nhiên gồm 9 chữ số từ 0 đến 9 không áp dụng bất cứ quy ước hiện hành nào. Họ tên là chuỗi ký tự in hoa với các từng thành phần họ, tên lót, tên đệm được lấy ngẫu nhiên từ tập dữ liệu các tên phổ biến. Trường ngày sinh được phát sinh ngẫu nhiên từ ngày 1/1/1960 đến ngày 31/12/2010 gồm hai loại định dạng chính cho dấu ngăn cách là gạch nối cho loại thẻ cấp từ năm 2010 và gạch chéo cho loại thẻ cấp trước năm 2010. Đối với nguyên quán và nơi cấp thẻ thì trường tỉnh thành phố được lấy từ cơ sở dữ liệu hành chính Việt Nam. Tương tự việc phát sinh địa chỉ thường trú, bên cạnh đó trường này cần thêm một số thông tin như số nhà, tên đường sẽ được lấy ngẫu nhiên trên tập dữ liệu có sẵn.

Về đặc trưng riêng của hai loại thẻ CMND cũ và mới. Đối với thẻ kiểu cũ định dạng chữ được dùng là Trixi Pro, trường số CMND có màu hồng đậm. Còn đối với thẻ kiểu mới, định dạng chữ được sử dụng là Palatino Linotype kết hợp với định dạng số CMND có định dạng Arial, các trường này đều đồng nhất là màu đen.

Các trường thông tin sau khi được ghi lên phôi sẽ được áp dụng ngẫu nhiên một lớp bộ lọc làm mờ sử dụng kỹ thuật Gaussian Blur [20] với kích thước bộ lọc là một số lẻ được phát sinh ngẫu nhiên trong khoảng 0.2% đến 0.3% kích thước chiều cao của phôi thẻ. Sau đó, tổng thể thẻ sẽ được áp dụng phương pháp hiệu chỉnh Gamma (Gamma Correction) [21] để thay đổi độ sáng ảnh.

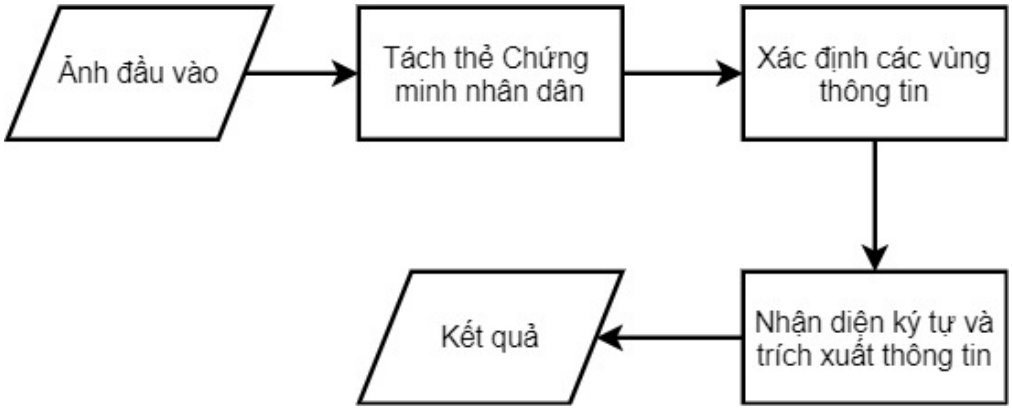
Hình 3.6 Áp dụng các bộ lọc lên ảnh

Hình 3.5 minh họa kết quả thay đổi ngẫu nhiên độ sáng của thẻ. Việc này giúp tập dữ liệu có độ đa dạng cao tương thích với ảnh được chụp từ người dùng trong các điều kiện ánh sáng thực tế.

CHƯƠNG 4: XÂY DỰNG GIẢI PHÁP

4.1. Tổng quan quy trình

Tổng quan cấu trúc xử lý của chương trình gồm 3 thành phần chính:



Hình 4.1 Cấu trúc xử lý của chương trình

- Tách thẻ Chứng minh nhân dân từ ảnh đầu vào
- Xác định các vùng thông tin quan trọng như số CMND, họ tên, địa chỉ,…
- Nhận diện các ký tự trên vùng thông tin và trích xuất thành thông tin

Mỗi thành phần đều sử dụng một mạng thần kinh riêng cho việc dự đoán và có thể áp dụng các kỹ thuật xử lý khác để tăng cường độ chính xác cho từng phần.

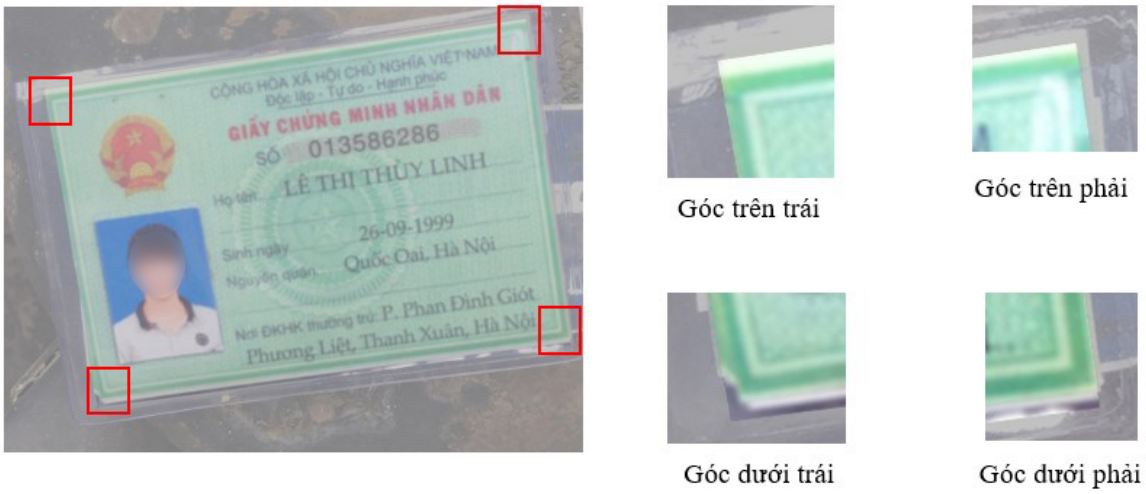
4.2. Tách thẻ Chứng minh nhân dân từ ảnh gốc

Ảnh Chứng minh nhân dân (CMND) đầu vào được người dùng chụp sẽ không tuân theo bất cứ quy tắc chuẩn nào nên kết quả ảnh chụp có thể dư sáng, thiếu sáng, thẻ CMND nằm vị trí giữa, hoặc lệch sang một bên nào đó của bức ảnh,… Do đó mục tiêu của phần này là xác định chính xác vị trí thẻ CMND và loại bỏ các vùng không liên quan trong ảnh người dùng chụp để có thể hỗ trợ tốt hơn cho phần xác định các vùng thông tin quan trọng. Một số hướng tiếp cận có thể có để giải quyết vấn đề này như dùng kết hợp các kỹ thuật xử lý ảnh để phát hiện đường viền cạnh,… nhưng độ hiệu quả sẽ không cao. Giải pháp được sử dụng trong luận văn này là dùng một mô hình học máy giải quyết vấn đề nhận diện vật thể.

Hình 4.2 Mô tả kết quả tách thẻ CMND ra khỏi nền xung quanh

4.2.1 Giải pháp tiếp cận

Mô hình nhận diện vận thể được dùng là mạng SSD kết hợp mô hình trích xuất đặc trưng MobileNet V2 để có thể cho ra kết quả tối ưu trong thời gian ngắn nhất. Đối tượng cần nhận diện trên ảnh là 4 gốc của thẻ CMND vì ảnh đầu vào được chụp ở khoảng cách gần nên có thể dễ dàng nhận diện được.



Hình 4.3 Đặc điểm nhận diện trên thẻ CMND

Ưu điểm lớn nhất của cách tiếp cận này là khi thẻ CMND trong ảnh nằm ở một góc nghiêng trong khoảng -90º đến 90º đều có thể đưa về ảnh nằm ngang thông thường bằng kỹ thuật biến đổi phối ảnh (Perspective Transform). Đây cũng là

nguyên nhân chính cho việc không thể chọn đối tượng nhận diện là thẻ CMND một cách tổng quát vì kết quả của mô hình SSD khi nhận dạng vật thể là hình chữ nhật, do đó khó xác định chính xác vị trí của thẻ CMND khi nằm nghiêng.

Nhược điểm của hướng tiếp cận này là khi thẻ CMND trong ảnh nằm nghiêng với góc lớn hơn 90º thì thẻ sẽ bị đảo ngược. Với giới hạn về thời gian thì luận văn chưa xử lý vấn đề này. Hướng giải quyết tốt nhất có thể là nhận diện thêm một đối tượng riêng biệt ở các vị trí không đối xứng như quốc hiệu, ảnh chân dung, … khi đó có thể xác định chính xác thẻ CMND ở bất kỳ vị trí nào. Bên cạnh đó việc nhận diện các góc của thẻ có thể cho ra kết quả với số góc nhỏ hơn 4. Đối với kết quả cho ra 3 góc thì có thể xử lý bằng các phép biến đổi giải tích để tìm góc còn lại. Đối với kết quả cho ra số lượng góc ít hơn thì không thể tiếp tục xử lý.

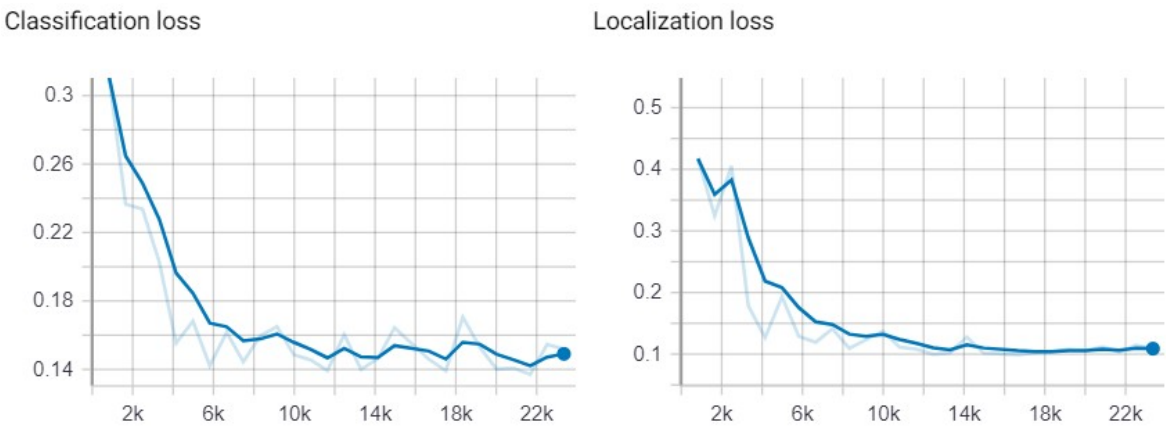
4.2.2 Chuẩn bị dữ liệu và huấn luyện

Tập dữ liệu huấn luyện gồm 3254 ảnh được chia thành 2 tập huấn luyện và kiểm thử lưu trữ ở dạng tập tin tfrecords (định dạng hỗ trợ bởi thử viện TensorFlow [22]) theo tỷ lệ tương ứng là 90% và 10%. Dữ liệu được lưu là thông tin về ảnh và nhãn các đối tượng gồm có tên ảnh, kích thước ảnh, dữ liệu về ảnh, loại ảnh, tên nhãn bằng chữ và số, và tọa độ đối tượng được gắn nhãn. Các nhãn được gán như bảng 4.1:

Bảng 4.1 Bảng gán nhãn các đối tượng nhận diện trên ảnh CMND

Đối tượng	**Nhãn**
Góc trên bên trái	C1
Góc trên bên phải	C2
Góc dưới bên trái	C3
Góc dưới bên phải	C4

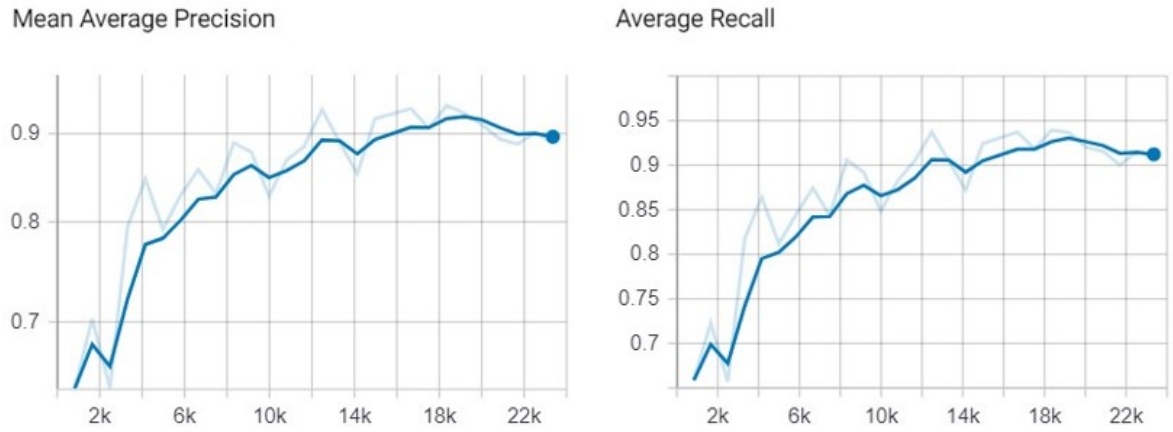
Mô hình được huấn luyện một lần với 23000 bước huấn luyện. Hàm mất mát được đánh giá dựa vào 2 tiêu chí là mất mát phân loại (Classification Loss) và mất mát cục bộ (Localization Loss) thu được kết quả như hình 4.4.



Hình 4.4 Giá trị hàm mất mát trong quá trình huấn luyện

4.2.3 Kết quả huấn luyện

Kết quả huấn luyện được đánh giá trên độ chính xác (Precision) và độ tái hiện (Recall), kết quả như hình 4.5:



Hình 4.5 Độ chính xác và độ tái hiện của mô hình

Từ kết quả cho thấy ở bước thứ 18500 mô hình đo được độ chính xác và độ tái hiện cao nhất nên sẽ lấy kết quả huấn luyện ở lần checkpoint gần nhất ở bước 19000.

4.3. Xác định các vùng thông tin quan trọng

Sau khi đã có được ảnh thẻ CMND một cách hoàn chỉnh thì việc xác định các vùng thông tin để đọc là rất quan trọng vì nó có thể giúp cải thiện độ chính xác cho phần trích xuất thông tin một cách hiệu quả. Các thông tin cần thiết để xác định như số CMND, họ tên, ngày sinh, địa chỉ thường trú, nguyên quán. Một cách tiếp cận đơn giản là lấy vùng hình chữ nhật đủ để chứa tất cả thông tin liên quan đến một trường nào đó nhưng hiệu quả mang lại không cao.

Hình 4.6 Cách nhận diện các vùng thông tin cơ bản

Hướng tiếp cận được áp dụng trong luận văn là sử dụng mô hình học máy nhận diện vật thể tương tự như đã áp dụng ở phần tách thẻ CMND.

4.3.1 Giải pháp tiếp cận

Tận dụng ưu thế của mô hình mạng SSD ở việc nhận dạng vật thể, trong phần này luận văn sử dụng mô hình trích xuất đặc trưng RetinaNet nhằm tăng độ chính xác khi nhận diện các đối tượng nhỏ như chữ in trên thẻ CMND. Mô hình sẽ nhận dạng và khoanh vùng các thông tin trên thẻ với độ dài từng từ đối với các thông tin như họ và tên, địa chỉ thường trú, nguyên quán. Đối với số CMND hay ngày sinh sẽ được nhận dạng riêng thành từng chuỗi với độ dài tương ứng.

Bảng 4.2 Đặc trưng riêng của các đối tượng

Đối tượng	**Đặc trưng**
Số Chứng minh nhân dân	Chữ số, màu đen hoặc đỏ thẫm, in trên nền gọn sóng màu đỏ
Họ và tên	Chữ in hoa, màu đen
Ngày/tháng/năm sinh	Chữ số (kèm ký tự phân cách), màu đen
Nguyên quán và Địa chỉ thường trú	Chữ thường, in hoa đầu mỗi từ, màu đen
Ngày cấp	Chữ số, màu đen, nền trắng
Nơi cấp	Chữ thường, in hoa đầu mỗi từ, màu đen, nền trắng

Việc chọn nhãn cho các đối tượng phụ thuộc vào sự khác nhau giữa các kiểu chữ và kích thước chữ của từng đối tượng. Đối với số CMND, họ tên, và ngày sinh đều có font và kích thước riêng biệt nên sẽ được đánh nhãn riêng cho từng loại. Đối với địa chị thường trú và nguyên quán thì không thể xác định được điểm khác biệt cho từng loại nên sẽ đánh một nhãn chung cho cả hai loại. Hình 4.7 minh họa chi tiết các gán nhãn trên.

Hình 4.7 Xác định nhãn cho các đối tượng

Kết quả đầu ra đạt được bao gồm tập hợp các ảnh chứa thông tin mỗi từ được sắp xếp theo thứ tự và phân nhóm theo các trường đối tượng trên thẻ CMND. Các

kết quả này được đưa sang phần tiếp theo để thực hiện nhận dạng và trích xuất thành các thông tin có ý nghĩa mà con người có thể đọc được.

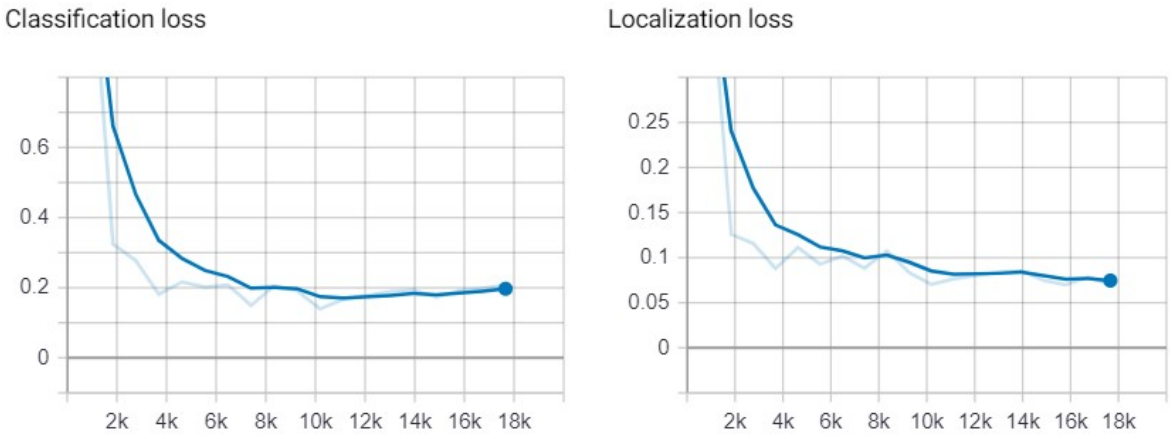
4.3.2 Chuẩn bị dữ liệu và huấn luyện

Tập dữ liệu gồm 180 ảnh được chia thành tập huấn luyện và tập kiểm thử với tỷ lệ tương ứng là 90:10. Tương tự quá trình chuẩn bị dữ liệu cho mô hình nhận dạng thẻ CMND, dữ liệu được lưu trữ ở dạng tập tin tfrecord bao gồm các thông tin như dữ liệu ảnh, tên ảnh, kích thước ảnh, tên các nhãn và tọa độ các đối tượng được gán nhãn. Các đối tượng được gán nhãn như bảng 4.1:

Bảng 4.3 Nhãn được gán cho các đối tượng

Đối tượng	**Tên nhãn**
Số Chứng minh nhân dân	Number
Họ và tên	Name
Ngày/tháng/năm sinh	Birthdate
Nguyên quán	Address
Địa chỉ thường trú	Address
Ngày cấp	Date
Nơi cấp	Province

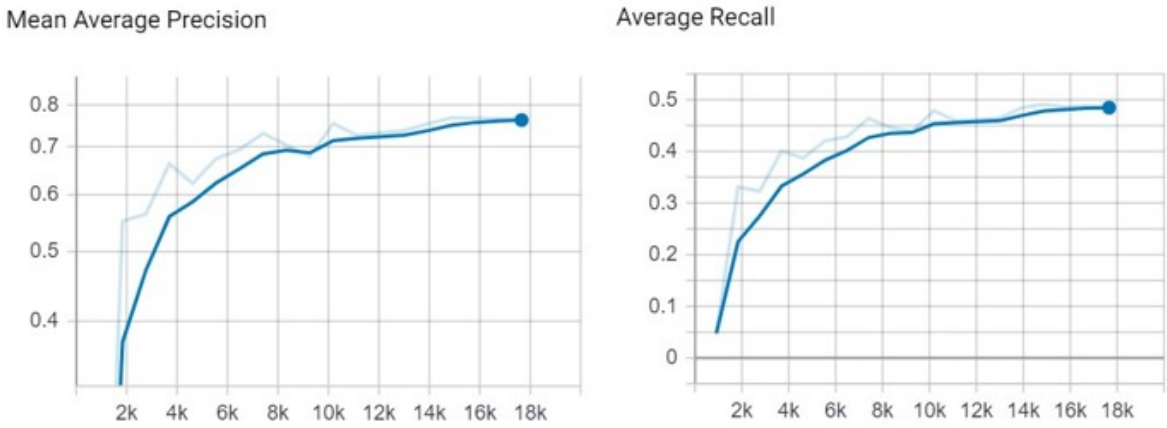
Tương tự phần 4.2, hàm mất mát của mô hình huấn luyện được đánh giá trên 2 tiêu chí chính là mất mát phân loại và mất mát cục bộ. Kết quả đo được sau hơn 17000 bước huấn luyện được mô tả như hình 4.8:



Hình 4.8 Giá trị hàm mất mát trong quá trình huấn luyện

4.3.3 Kết quả huấn luyện

Kết quả huấn luyện dựa trên độ chính xác và độ tái hiện của mô hình thu được như hình 4.9 :

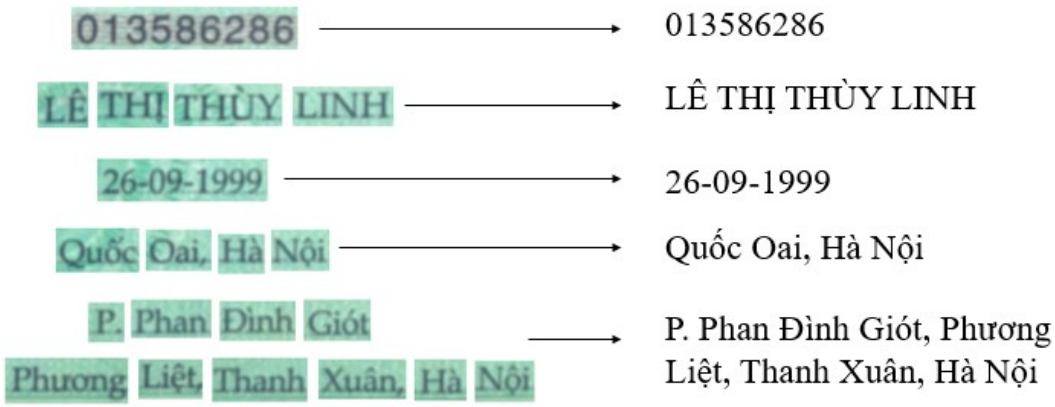


Hình 4.9 Kết quả huấn luyện mô hình nhận đạng vùng thông tin

Từ kết quả ở hình 4.9 cho thấy mô hình đã đạt ngưỡng độ chính xác cao nhất sau 18000 bước huấn luyện.

4.4. Nhận diện ký tự và trích xuất thông tin

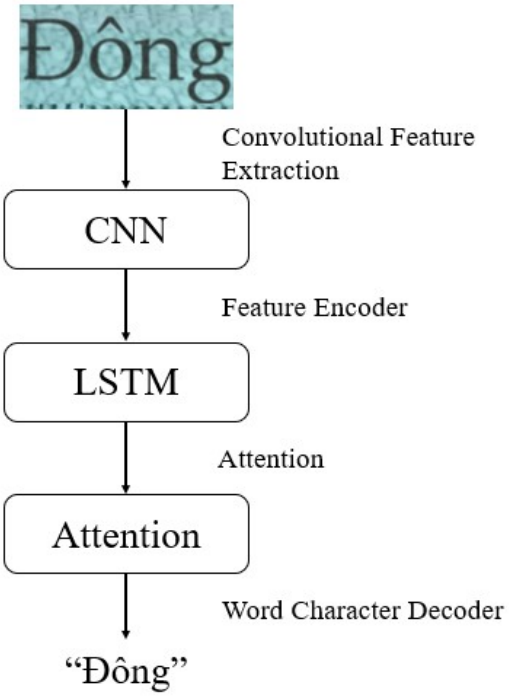
Sau khi đã có được các chuỗi ảnh chứa thông tin từ phần 4.3. Các ảnh này được đưa vào một mô hình học máy để thực hiện quá trình nhận dạng ký tự. Các kết quả nhận diện sẽ được sắp xếp theo thứ tự và đưa ra thông tin cuối cùng.



Hình 4.10 Mô tả kết quả trích xuất thông tin

4.4.1 Giải pháp tiếp cận

Trong luận văn này sử dụng một mô hình sử dụng kỹ thuật chú ý (Attention-based) để nhận diện ký tự quang học (OCR) của tác giả Qi Guo và Yuntian Deng [23]. Ảnh chứa từ khóa đầu vào được trích xuất đặc trưng từ các lớp CNN thành véc-tơ đặc trưng, véc-tơ đặc trưng này được dùng làm đầu vào cho mô hình Encoder. Kết quả thu được từ các cell được tổng hợp ở lớp Attention và giải mã thành ký tự.



Hình 4.11 Cấu trúc mô hình mạng nhận diện ký tự

4.4.2 Chuẩn bị dữ liệu và huấn luyện

Trong giới hạn luận văn, dữ liệu huấn luyện được tạo bằng chương trình Python dựa trên các đặc trưng tiêu chuẩn của các loại thẻ CMND hiện đang được sử dụng ở thời điểm hiện tại. Trong đó chia thành 2 loại chính là thẻ CMND kiểu cũ (thẻ được cấp từ trước năm 2010) và thẻ CMND kiểu mới (thẻ được cấp từ năm 2010 trở về sau).

CMND kiểu cũ CMND kiểu mới

Hình 4.12 So sánh chứng minh nhân dân kiểu cũ và mới

Bảng 4.4 So sánh đặc điểm thẻ CMND cũ và mới

	CMND kiểu cũ	**CMND kiểu mới**
Kiểu chữ	Trixi Pro	Palatino Linotype
Kiểu số CMND	Trixi Pro	Arial
Màu chữ	Đen xám	Đen
Màu số CMND	Cánh sen đậm	Đen

Mục tiêu của mô hình là nhận diện ký tự tiếng Việt nên bảng ký tự được đưa vào huấn luyện bao gồm 10 ký tự số 0 ... 9, 186 ký tự chữ cái tiếng Việt a, ă, â, ... và 5 ký tự phân cách gồm dấu chấm, phẩy, gạch chéo, gạch ngang và nháy đơn.

Tập huấn luyện có kích thước tối đa 30000 ảnh chứa các từ, chữ cái, số, chuỗi số được cắt từ ảnh CMND. Cụ thể bao gồm 10% ảnh số CMND, 32% ảnh họ tên,

11% ảnh ngày sinh, 21% ảnh địa chỉ và nguyên quán, còn lại là các ký tự và chữ số đơn lẻ.

Trước khi đưa vào mô hình huấn luyện, các ảnh được thay đổi kích thước về chuẩn kích thước tối đa là 160 x 60. Điều này làm giảm thời gian huấn luyện một cách đáng kể tuy nhiên vẫn có nguy cơ gây mất thông tin dữ liệu của ảnh.

Mô hình được thí nghiệm huấn luyện 3 lần với kích thước tập huấn luyện lần lượt là 10000 ảnh, 20000 ảnh, và 30000 ảnh với 20000 bước huấn luyện. Thông số huấn luyện được cài đặt như sau:

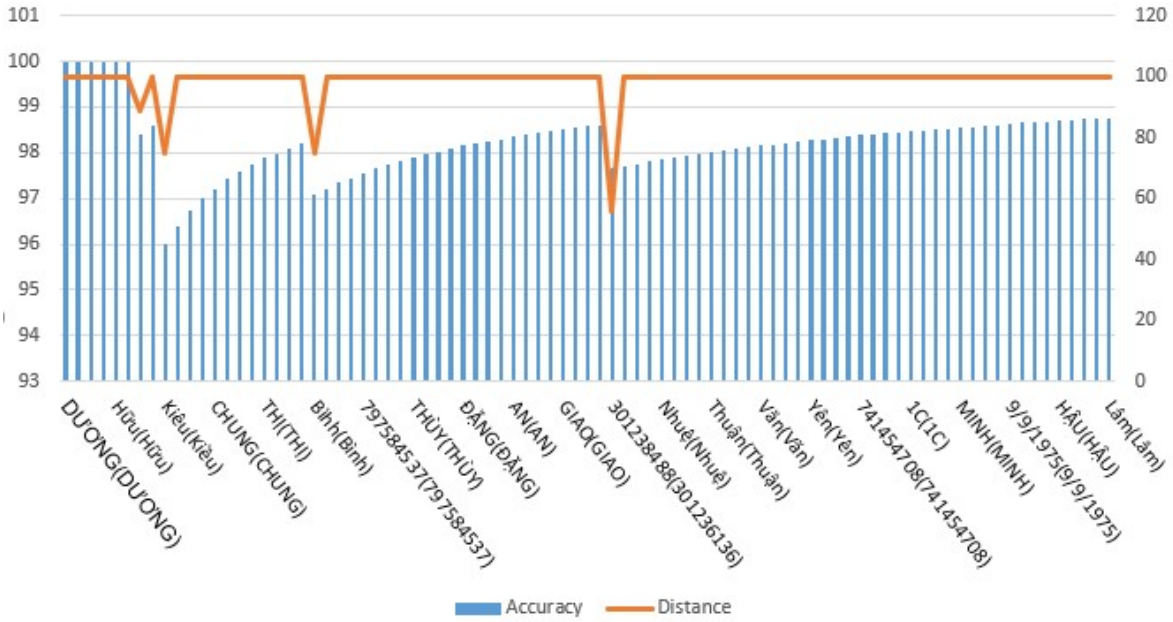
- *steps-per-checkpoint*: 2000
- *batch-size*: 16
- *num-epoch*: $\frac{Số\ bước\ huấn\ luyện}{\frac{Kích\ thước\ tập\ huấn\ luyện}{Batch-size}}$
- *max-prediction*: 12
- *initial-learning-rate*: 1.0

Trong đó *steps-per-checkpoint* là lưu checkpoint sau bao nhiêu bước huấn luyện; *batch-size* là số lượng dữ liệu (ảnh) được đưa vào một bước huấn luyện; *num-epoch* là số lần lặp sau khi huấn luận toàn bộ dữ liệu 1 lần; *max-prediction* là độ dài chuỗi tối đa trong tập dữ liệu huấn luyện; *initial-learning-rate* là tốc độ học khởi tạo, trong luận văn dùng thuật toán tối ưu hóa hội tụ là Adadelta.

Hình 4.13 Giá trị độ hỗn độn trong quá trình huấn luyện

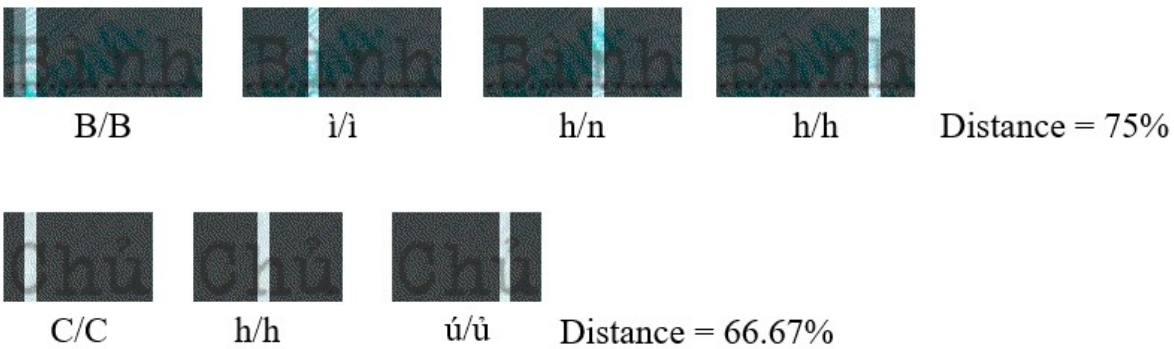
4.4.3 Kết quả huấn luyện

Sau quá trình huấn luyện, mô hình được kiểm thử trên tập dữ liệu gồm 100 ảnh chứa từ khóa các loại đạt được độ chính xác trung bình là 98.61%, chi tiết được mô tả ở hình 4.14:



Hình 4.14 Kết quả kiểm thử mô hình OCR trên tập dữ liệu 100 ảnh

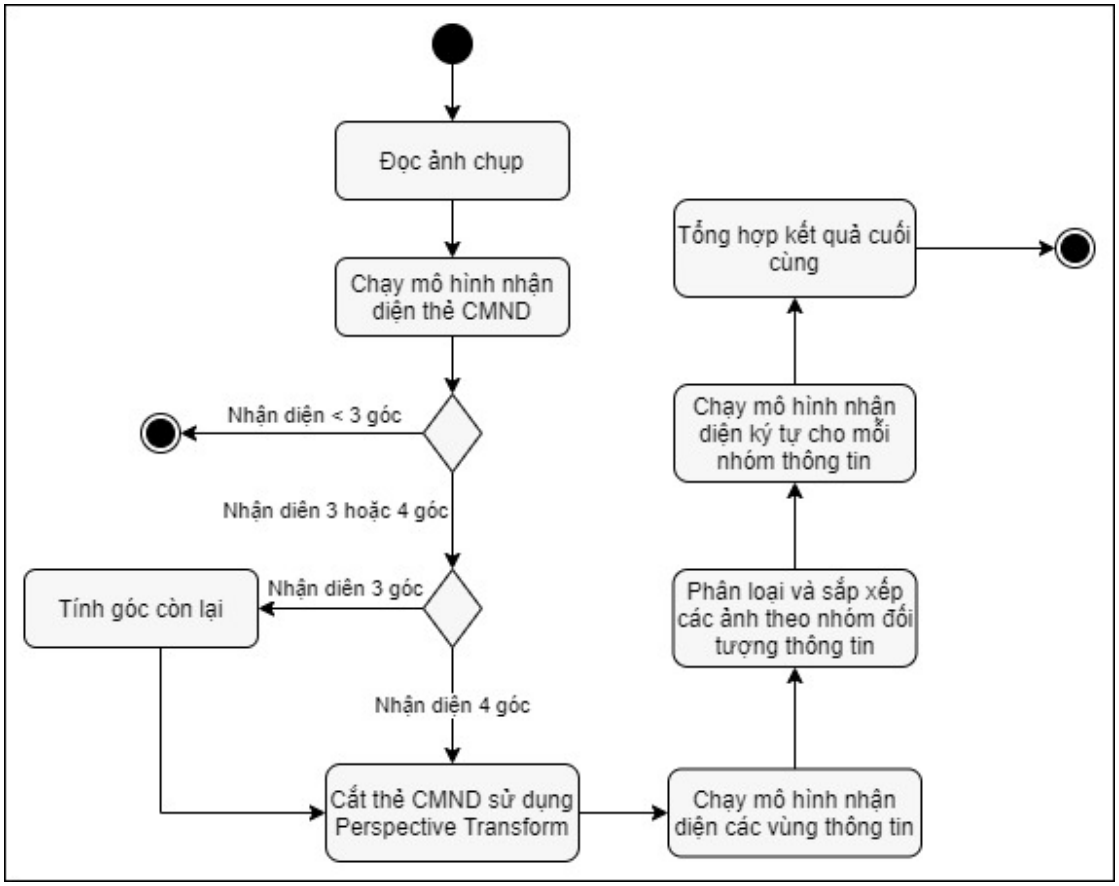
Độ đo được sử đụng để đánh giá độ chính xác của một từ là Khoảng cách Levenshtein (Levenshtein distance [24]). Từ kết quả ở hình 4.14 cho thấy 98% các chuỗi được nhận dạng đúng, bên cạnh đó vẫn còn một số chuỗi được nhận dạng chưa chính xác như “Bìhh”, “Kiêu”, “Chú”…



Hình 4.15 Một số chuỗi được dự đoán sai

4.5. Chương trình xử lý đầu cuối

Với mục đích trích xuất được thông tin từ ảnh chụp CMND, 3 mô hình huấn luyện trên được đưa vào một chương trình hoàn chỉnh có thể xử lý từ bước đầu tiên là nhận ảnh chụp tới bước đưa ra thông tin trích xuất một cách tối ưu nhất. Cấu trúc chương trình được mô tả như hình 4.16:



Hình 4.16 Cấu trúc chương trình

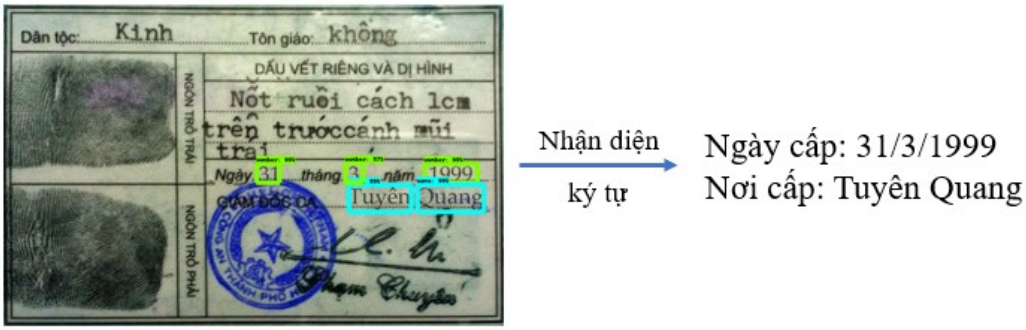
Như đã đề cập ở phần 4.2, mô hình nhận diện thẻ CMND chỉ dự đoán được 1 hoặc 2 góc thì chương trình không thể xử lý tiếp tục được. Việc cắt thẻ sử dụng kỹ thuật Perspective Transform sẽ đưa thẻ về một dạng thống nhất để hỗ trợ cho các quá trình xử lý tiếp theo. Ở bước sắp xếp các ảnh thông tin theo nhóm đối tượng, để loại bỏ các vùng thông tin mà mô hình nhận diện sai hoặc dư thừa, luận văn sử dụng một số vị trí cố định trên thẻ CMND làm các đường giới hạn như đường nằm giữa

vùng Ngày sinh và vùng Nguyên quán, … do đó làm tối ưu hóa kết quả dự đoán của mô hình.

Mô hình nhận diện ký tự được áp dụng sau cùng cho từng nhóm đối tượng và đóng vai trò quan trọng cho kết quả trích xuất thông tin sau cùng của chương trình. Các thông tin được tổng hợp và xử lý nhiễu như loại bỏ dấu câu thừa, chỉnh sửa lỗi chính tả, … Kết quả đầu ra của mỗi mô hình được mô tả như hình 4.17:

Hình 4.17 Kết quả dự đoán các mô hình trong chương trình

Từ kết quả hình 4.17 cho thấy phần lớn các thông tin quan trọng được trích xuất chính xác ngoại trừ một số lỗi như thiếu dấu câu ở địa chỉ thường trú, ngày sinh thiếu dấu gạch nối giữa tháng và năm sinh. Tại bước này có thể xử lý các thông tin nhiễu trên bằng việc xử dụng một số thuật toán đơn giản chỉnh sửa các lỗi liên quan đến cú pháp, chính tả, định dạng ở các trường họ tên, ngày sinh, hoặc địa chỉ nguyên quán và thường trú.



Hình 4.18 Kết quả nhận diện ký tự mặt sau

Hình 4.18 cho kết quả nhận diện tương tự đối với mặt sau. Thời gian ngày, tháng, năm và nơi cấp được mô hình nhận diện chính xác do đó giảm thiểu sai sót cho mô hình nhận diện ký tự.

CHƯƠNG 5: THỰC NGHIỆM VÀ ĐÁNH GIÁ

5.1. Mô hình nhận diện thẻ CMND

Mô hình được thực nghiệm trên tập 131 ảnh chụp CMND sưu tầm được với các loại chất lượng và điều kiện ảnh khác nhau. Kết quả dự đoán các góc được mô tả như bảng 5.1:

Bảng 5.1 Kết quả dự đoán các góc của mô hình nhận diện thẻ CMND

Số lượng góc / Kết quả	1 góc	2 góc	3 góc	4 góc
Số lượng (ảnh)	11	25	53	42
Tỉ lệ (%)	8.4	19	40.5	32.1

Từ kết quả ta có thể thấy được tỉ lệ nhận diện thẻ CMND thành công với kết quả từ 3 đến 4 góc chiếm 72.6%. Đây là kết quả chấp nhận được khi áp dụng ở môi trường thực tế. Bên cạnh đó vẫn có thể tăng tỷ lệ nhận diện thành công khi huấn luyện mô hình với dữ liệu thật.

5.2. Mô hình nhận diện vùng thông tin

Mô hình nhận diện các vùng thông tin được thực nhiệm trên 100 ảnh CMND được tạo từ chương trình phát sinh và được giữ nguyên kích thước ảnh trong quá trình thực nghiệm. Để đánh giá kết quả của mô hình, luận văn sử dụng độ đo Precision, Recall và F-measure [25] được tính như sau:

$$Precision = \frac{Số\ nhận\ diện\ đúng}{Số\ nhận\ diện\ đúng + Số\ nhận\ diện\ sai}$$

$$Recall = \frac{Số\ nhận\ diện\ đúng}{Số\ nhận\ diện\ đúng + Số\ không\ nhận\ diện}$$

$$F - measure = \frac{2 * Precision * Recall}{Precision + Recall}$$

Để xác định mô hình nhận diện đúng một ảnh dựa sẽ dựa trên tiêu chí độ đo IoU (Intersection over Union) [26] không nhỏ hơn 70%, đây là giá trị cần thiết cho mô hình nhận diện ký tự có thể dự đoán chính xác từ khóa. Kết quả được thống kê ở bảng 5.2:

Bảng 5.2 Kết quả thực nghiệm mô hình nhận diện vùng thông tin

Trường dữ liệu	N_D	N_W	N_U	Precision (%)	Recall (%)	F-measure (%)
Số CMND	91	0	9	100	91	95.3
Họ tên	276	65	4	80.9	98.5	88.8
Ngày sinh	100	0	0	100	100	100
Địa chỉ	879	90	2	90.7	99.7	95

Với N_D là số trường hợp nhận diện chính xác, N_W là số trường hợp nhận diện không chính xác và N_U là số trường hợp không nhận diện được. Từ kết quả F-measure cho thấy chất lượng mô hình nhận diện khá tốt tuy vẫn cần tăng cường độ chính xác ở đối tượng họ tên.

5.3. Mô hình nhận diện ký tự

Độ chính xác của mô hình khi nhận diện các trường thông tin được tính trên độ đo Levenshtein [24] được mô tả như sau:

$$lev_{a,b}(i,j) = \begin{cases} \max(i,j) & if\ \min(i,j) = 0, \\ min \begin{cases} lev_{a,b}(i-1,j) + 1 \\ lev_{a,b}(i,j-1) + 1 \\ lev_{a,b}(i-1,j-1) + 1\ (a_i \neq b_j) \end{cases} & otherwise \end{cases}$$

Với a, b là hai chuỗi có độ dài lần lượt là i, j; $lev_{a,b}(i,j)$ là i ký tự đầu của chuỗi a và j ký tự đầu của chuỗi b; và $lev_{a,b}$ là khoảng cách Levenshtein của hai chuỗi.

Để thấy rõ điểm mạnh của mô hình được đề xuất, một thực nghiệm được thực hiện đánh giá kết quả dự đoán ký tự trên các trường thông tin của thẻ CMND giữa mô hình được đề xuất với mô hình Tesseract OCR do Google phát triển sử dụng gói

ngôn ngữ tiếng Việt. Đối với mô hình Tesseract, các ảnh đầu vào trước khi được nhận diện sẽ trải qua một số bước tiền xử lý ảnh như làm mịn ảnh (Image Smoothing) để loại bỏ các chi tiết thừa ở phần nền và phân ngưỡng ảnh (Image Thresholding) để làm nổi bật chi tiết từ khóa so với ảnh nền.

Dữ liệu thử nghiệm được sử dụng với số lượng 131 ảnh CMND mặt trước và 107 ảnh CMND mặt sau. Kết kết quả kiểm thử được mô tả như bảng 5.3:

Bảng 5.3 Kết quả so sánh độ chính xác của mô hình OCR Tesseract và mô hình đề xuất

	Độ chính xác	
	OCR Tesseract	**Mô hình đề xuất**
Số CMND	82.4%	94.8%
Họ tên	69.7%	87.7%
Ngày sinh	85.7%	95.4%
Nguyên quán	72.3%	85.5%
Địa chỉ thường trú	79%	85.3%
Ngày cấp	85.1%	97.5%
Nơi cấp	62.4%	70.6%

Nhìn chung thì mô hình đề xuất cho kết quả tốt hơn so với mô hình Tesseract. Các trường thông tin liên quan đến số như số CMND, ngày sinh, ngày cấp cho kết quả chính xác cao nhất. Các trường ký tự còn lại cho kết quả chính xác thấp hơn. Nguyên nhân dẫn đến sự chênh lệch này có thể do mô hình nhận diện vùng thông tin xác định xót đối tượng trên cùng một trường thông tin. Bên cạnh đó sự đa dạng dữ liệu huấn luyện cho mô hình nhận diện ký tự cũng có ảnh hưởng trực tiếp đến kết quả này.

Để so sánh và xác định rõ hơn, bảng 5.4 liệt kê một số trường hợp khó nhận diện khi ảnh chụp trong điều kiện thiếu sáng, thẻ bị mờ, hoặc các thẻ kiểu cũ.

Bảng 5.4 Kết quả và độ chính xác của mô hình Tesseract và mô hình đề xuất trên tập các mẫu khó nhận diện

Chuỗi	OCR Tesseract		Mô hình đề xuất	
	Dự đoán	**Độ chính xác**	**Dự đoán**	**Độ chính xác**
TẠ LÊ ĐIỀN 444084559	TA ĐỂ ĐIỆN 114094559	63.3%	TẠ LÊ ĐIỀN 444084559	100%
CHUNG NGỌC HỒNG ANH 060431313	GHUNG NGức HÔNa AaKH 6484141212:	37.1%	CHUNG NGỌC HÙNG AANH 060631713	83.6%
NGUYỄN KIỆN TƯỜNG 870010346	NGUYEN KIỆN TƯỜNG 870010346	97%	NGUYỄN KIỆN TƯỜNG 870010346	100%
NGUYỄN VIẾT DUY 799618020	NGUYÊN Yt#n UY 20962802	57.7%	TGUYỄN VIỆT DU7 399618020	84.4%
CHUNG PHƯỚC HỘI 733271264	CHƯNG PHƯỚC HỘI 783973264	80%	CHUNG THƯỚC HỘI 733271264	96.6%
LONG ĐÔNG ĐẠI 824715473	LÔNG ĐÔNG ĐẠI 824115475	85%	L0NG ĐÔNG ĐẠI 824715473	96.1%

Với mô hình Tesseract, phần lớn kết quả dự đoán trường số CMND đều có sai sót. Điều này xảy ra do phần số CMND được in trên phần gợn sóng màu đỏ của thẻ gây nhiễu cho mô hình nhận diện.

Mô hình đề xuất cho kết quả dự đoán đạt độ chính xác tương đối cao. Tuy vẫn có một số sai sót nhất định nhưng đối với các ảnh chụp trong điều kiện chưa tốt hay tình trạng thẻ CMND xấu thì kết quả này có thể chấp nhận được.

CHƯƠNG 6: KẾT LUẬN

Luận văn là sản phẩm của quá trình làm việc và nghiên cứu của nhóm sinh viên để đưa ra một cách giải quyết cho bài toán nhận diện và trích xuất thông tin từ các loại giấy tờ xác minh danh tính như căn cước công dân, bằng lái xe, hộ chiếu,…

Đối tượng nghiên cứu chính trong luận văn này là nhận diện và trích xuất thông tin từ thẻ chứng minh nhân dân bằng cách sử dụng các mô hình học máy để xác định vị trí thẻ, khoanh vùng thông tin cần trích xuất, phân tách từ khóa thành từng từ hoặc cụm và nhận diện ký tự quang học. Việc xác định vị trí thẻ được mô hình dự đoán dựa trên đối tượng bốn góc của thẻ CMND. Kết quả dự đoán sẽ được tính toán để thực hiện cắt hoàn chỉnh thẻ CMND làm đầu vào cho mô hình khoanh vùng và phân tách từ khóa. Sau khi các thông tin được đóng gói thành các chuỗi ảnh chứa từ khóa, mô hình nhận diện ký tự sẽ đọc và đưa ra các chuỗi thông tin dạng văn bản và cho ra kết quả cuối cùng.

Luận văn đã đạt được nhiều kết quả khả quan cũng như tiềm năng áp dụng ở môi trường thực tế. Quá trình thực hiện luận văn mang lại cho sinh viên những kiến thức và kỹ thuật trong lĩnh vực khoa học máy tính như các khái niệm mạng thần kinh tích chập, kỹ thuật chú ý, mô hình chuyển đổi chuỗi sang chuỗi, các mô hình học máy nhận diện vật thể, nhận diện ký tự,… được áp dụng trong thực tế. Bên cạnh đó, việc sử dụng ngôn ngữ Python cũng như các thư viện hỗ trợ như Numpy, OpenCV, Tensorflow là kỹ năng không thể thiếu trong quá trình tạo dữ liệu và huấn luyện các mô hình.

Tuy đạt kết quả khá tốt trong môi trường thử nghiệm, luận văn còn một số điểm hạn chế cần khắc phục khi áp dụng ở môi trường thực tế. Một trong số đó là thời gian xử lý của chương trình và thời gian dự đoán của các mô hình chưa được tối ưu. Tiếp đến là mô hình xác định vị trí thẻ CMND nhận diện chưa chính xác nếu ảnh chụp ngược chiều thẻ. Bên cạnh đó, kết quả nhận diện ký tự cho các thông tin quan trọng như số CMND và ngày sinh vẫn chưa đạt độ chính xác tối đa.

Như đã đề cập ở chương Mở đầu, xã hội ngày càng phát triển theo xu hướng số hóa – công nghệ hóa sẽ làm phát sinh và tăng nhu cầu giải quyết các bài toán về nhận diện ký tự quang học, do đó luận văn cũng có rất nhiều hướng phát triển về sau. Trong phạm vi ngắn hạn, việc cấp thiết cần làm là giải quyết các hạn chế còn tồn tại của luận văn như xác định vị trí thẻ CMND ở các phương chiều khác nhau, tăng cường độ chính xác cho mô hình nhận diện ký tự và tối ưu hóa quy trình xử lý cũng như các mô hình nhận diện để có thể cho kết quả trong thời gian ngắn nhất. Trong phạm vi dài hạn, các mô hình nhận diện có thể hoạt động trên không chỉ riêng thẻ CMND, mà còn có thể hoạt động được trên thẻ Căn cước công dân, hoặc các thẻ tương tự khác như Giấy phép lãi xe, Hộ chiếu,…

TÀI LIỆU THAM KHẢO

[1] Daniel Graupe. “Principles of Artificial Neural Networks”. World Scientific, 2013. Chapter 2.

[2] Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. May 2016.

[3] Jianxin Wu. Indroduce to Convolutional Neural Networks, May 1 2017. Chapter 6: The convolution layer.

[4] Jianxin Wu. Indroduce to Convolutional Neural Networks, May 1 2017. Chapter 7: The polling layer.

[5] Yoshua Bengio, Patrice Simard, Paolo Frasconi. Learning long-Term Dependencies with Gradient Descent is Difficult – IEEE Transactions on Neural Networks, Vol 5, No. 2. March 1994.

[6] Wojciech Zaremba, Ilya Sutskever, Oriol Vinyals. Recurent Neural Networks Regularization, 2015. Chapter 2: Related Work.

[7] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szedegy, Scott Reed, Cheng-Yang Fu, Alexanger C. Berg. SSD: Single Shot Multibox Detector. 29 December 2016.

[8] Christian Szedegy, Scott Reed, Dumitru Erhan, Dragomir Anguelov, Sergey Ioffe. Scalable High Quality Object Detection. 9 December 2015.

[9] Karen Simonyan, Andrew Zisserman. Very Deep Convolutional Networks for Large-scale Image Recognition. 10 April 2015.

[10] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Macro Andreetto, Hartwig Adam. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. 17 Apr 2017.

[11] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, Liang-Chieh Chen. MobileNetV2: Inverted Residuals and Linear Bottlenecks. 21 Mar 2019.

[12] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, Piotr Dollar. Focal Loss for Dense Object Detection. 7 Feb 2018.

[13] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and ´ S. Belongie. Feature pyramid networks for object detection. In CVPR, 2017.

[14] Ilya Sutskever, Oriol Vinyals, Quoc V. Le. Sequence to Sequence Learning with Neural Networks. Dec 2014.

[15] Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. May 2016.

[16] Chung-Cheng Chiu, Tara N. Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J. Weiss, Kanishka Rao, Ekaterina Gonina, Navdeep Jaitly, Bo Li, Jan Chorowski, Michiel Bacchiani. State-of-the-art Speech Recognition With Sequence-to-Sequence Models. Feb 2018.

[17] Chandra Khatri, Gyanit Singh, Nish Parikh. Abstractive and Extractive Text Summarization using Document Context Vector and Recurrent Neural Networks. Jul 2018.

[18] Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason J. Corso, Jianfeng Gao. Unified Vision-Language Pre-Training for Image Captioning and VQA. Dec 2019.

[19] Devendra Kumar Sahu, Mohak Sukhwani. Sequence to Sequence Learning for Optical Character Recognition. Dec 2015.

[20] Estevão S. Gedraite, Murielle Hadad. Investigation on the effect of a Gaussian Blur in image filtering and segmentation. Oct 2011.

[21] Gang Cao, Lihui Huang, Huawei Tian, Xianglin Huang, Yongbin Wang, Ruicong Zhi. Contrast Enhancement of Brightness-Distorted Images by Improved Adaptive Gamma Correction. Sep 2017.

[22] TFRecord and tf.Example. URL: https://www.tensorflow.org/tutorials/load_data/tfrecord. Accessed: 06/23/2020.

[23] Visual Attention based OCR. URL: https://github.com/da03/Attention-OCR. Accessed: 06/01/2020.

[24] Dan Hirschberg. Serial Computations of Levenshtein Distances. 1997.

[25] David Martin Ward Powers. Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation. Jan 2008.

[26] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, Silvio Savarese. Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression. 15 Apr 2019.