BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC THĂNG LONG

KHÓA LUẬN TỐT NGHIỆP

ỨNG DỤNG TRÍ TUỆ NHÂN TẠO PHÁT HIỆN BỆNH VÕNG MẠC ĐÁI THÁO ĐƯỜNG

GIẢNG VIÊN HƯỚNG DẪN: TS. NGUYỄN THỊ HUYỀN CHÂU

SINH VIÊN THỰC HIỆN: A40670 - NGUYỄN THỊ ÁNH

A40405 - BÙI HỮU HUẤN

NGÀNH: TRÍ TUỆ NHÂN TẠO

HÀ NỘI – 2024

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC THĂNG LONG

KHÓA LUẬN TỐT NGHIỆP

ỨNG DỤNG TRÍ TUỆ NHÂN TẠO PHÁT HIỆN BỆNH VÕNG MẠC ĐÁI THÁO ĐƯỜNG

GIẢNG VIÊN HƯỚNG DẪN: TS. NGUYỄN THỊ HUYỀN CHÂU

SINH VIÊN THỰC HIỆN: A40670 - NGUYỄN THỊ ÁNH

A40405 - BÙI HỮU HUẤN

NGÀNH: TRÍ TUỆ NHÂN TẠO

HÀ NỘI – 2024

THANG LONG UNIVERSITY

GRADUATE DISSERTATION

APPLICATION OF ARTIFICIAL INTELLIGENCE TO DETECTE DIABETIC RETINOPATHY

SUPERVISION: TS. NGUYỄN THỊ HUYỀN CHÂU

STUDENT: A40670 - NGUYỄN THỊ ÁNH

A40405 - BÙI HỮU HUẤN

MAJOR: TRÍ TUỆ NHÂN TẠO

HÀ NỘI – 2024

LỜI CẢM ƠN

Đầu tiên, chúng em xin gửi lời cảm ơn tới toàn thể giảng viên trường Đại học Thăng Long, những người thầy, người cô đã trực tiếp giảng dạy và truyền đạt những kiến thức suốt 4 năm chúng em học tại trường. Đó chính là nền tảng và hành trang vô cùng quý giá giúp chúng em vững bước trên hành trình xây dựng sự nghiệp tương lai của mình.

Chúng em cũng xin gửi lời cảm ơn chân thành đến cô Nguyễn Thị Huyền Châu, cô đã tận tình hướng dẫn để chúng em có thể hoàn thành khóa luận tốt nghiệp một cách tốt nhất.

Để có được kết quả như ngày hôm nay, chúng em rất biết ơn các thầy cô trong phòng thí nghiệm Trí tuệ nhân tạo trường Đại học Thăng Long, gia đình và bạn bè đã động viên, khích lệ, tạo mọi điều kiện thuận lợi nhất cho chúng tôi trong suốt quá trình học tập cũng như quá trình thực hiện khóa luận tốt nghiệp này.

Vì thời gian nghiên cứu và kiến thức còn hạn chế, nên chúng em không thể tránh khỏi những thiếu sót trong bài luận văn của mình. Chúng em cũng rất mong nhận được những ý kiến đóng góp của các thầy cô để hoàn thiện khóa luận tốt nghiệp.

SINH VIÊN

Nguyễn Thị Ánh

Bùi Hữu Huấn

LỜI CAM ĐOAN

Chúng tôi xin cam đoan đề tài này được thực hiện dựa trên lý thuyết, thuật toán xử lý dữ liệu, thu thập dữ liệu và các mô hình được trình bày trong khoá luận là do chúng tôi thực hiện dưới sự hướng dẫn của TS. Nguyễn Thị Huyền Châu. Mọi tài liệu, các bài báo và công cụ của các tác giả khác được sử dụng đều có trích dẫn tường minh về nguồn gốc và tác giả trong danh sách tài liệu tham khảo. Chúng tôi xin chịu hoàn toàn trách nhiệm về lời cam đoan này!

Hà Nội, ngày tháng năm 2024

SINH VIÊN

Nguyễn Thị Ánh

Bùi Hữu Huấn

MỤC LỤC

DANH MỤC CÁC TỪ VIẾT TẮT

Từ viết tắt	Ý nghĩa
CNN	Convolutional Neural Network
CV	Cross validation
DR	Diabetic retinopathy (bệnh võng mạc đái tháo đường)

DANH MỤC CÁC HÌNH VẼ

DANH MỤC CÁC BẢNG

DANH MỤC THUẬT NGỮ

Thuật ngữ trong y tế	
Thuật ngữ	**Ý nghĩa**
Chẩn đoán	Quá trình xác định hoặc đưa ra một đánh giá về tình trạng sức khỏe của một người bệnh dựa trên các triệu chứng, thông tin lâm sàng và kết quả các bài kiểm tra y tế. Chẩn đoán giúp xác định loại bệnh, đặc điểm của bệnh và đưa ra kế hoạch điều trị phù hợp.
Đái tháo đường	Một bệnh lý mà cơ thể không thể điều chỉnh mức đường trong máu một cách hiệu quả, dẫn đến tình trạng mức đường trong máu tăng cao.
Bệnh võng mạc đái tháo đường	Một biến chứng của bệnh đái tháo đường ảnh hưởng đến võng mạc, gây tổn thương và suy giảm thị lực.
Võng mạc	Một mô mắt nhạy sáng nằm ở phía sau mắt, chứa các tế bào nhận thức ánh sáng và chuyển đổi nó thành tín hiệu điện để gửi đến não.
Ảnh chụp võng mạc	là thuật ngữ y tế chỉ việc sử dụng các thiết bị và phương pháp hình ảnh để chụp và ghi lại hình ảnh của võng mạc mắt.
Phù hoàng điểm	Điểm vàng bị phù nề do ứ dịch

Thuật ngữ trong công nghệ thông tin	
Thuật ngữ	**Ý nghĩa**
Trí tuệ nhân tạo	Trí tuệ do con người lập trình tạo nên với mục tiêu giúp máy tính có thể tự động hóa các hành vi thông minh như con người.
Dữ liệu	Một tập hợp các dữ kiện, chẳng hạn như số, từ, hình ảnh, nhằm đo lường, quan sát hoặc chỉ là mô tả về sự vật
Trọng số	Một loại giá trị được sử dụng để tính toán, trong đó mỗi giá trị để tính toán đều mang mức độ quan trọng khác nhau
Siêu tham số	Những giá trị tham số có thể điều chỉnh quá trình huấn luyện của mô hình
Bias	Độ lệch, biểu thị sự chênh lệch giữa giá trị trung bình mà mô hình dự đoán và giá trị thực tế của dữ liệu
Tập huấn luyện	Tập dữ liệu sử dụng để huấn luyện mô hình
Tập kiểm thử	Tập dữ liệu sử dụng để kiểm tra độ chính xác của mô hình
Tập xác thực	Tập dữ liệu sử dụng để đánh giá mô hình khi đang huấn luyện
Tiền huấn luyện	Mô hình đã được huấn luyện trước đó với các phương pháp hiện đại giúp giảm công sức huấn luyện mô hình từ đầu

MỞ ĐẦU

Trong chương đầu tiên, tài liệu sẽ mô tả các vấn đề liên quan tới đề tài. Sau đó, tài liệu sẽ mô tả các mục tiêu nghiên cứu, phương pháp nghiên cứu, đối tượng nghiên cứu, phạm vi nghiên cứu. Các phần này sẽ được trình bày cụ thể hơn dưới dây.

1. Đặt vấn đề

Theo Thứ trưởng Bộ Y tế Nguyễn Thị Liên Hương, hiện nay, tại Việt Nam có khoảng gần 5 triệu người đang mắc bệnh đái tháo đường thì hơn 55% bệnh nhân mắc bệnh đái tháo đường đã có biến chứng [1]. Bệnh võng mạc đái tháo đường là một trong những biến chứng nguy hiểm của bệnh đái tháo đường, có khoảng 20% những người mắc tiểu đường có biến chứng ở mắt với các mức độ khác nhau [2]. Nếu không được phát hiện và điều trị kịp thời có thể dẫn đến mù lòa. Việc chẩn đoán các mức độ của bệnh thực hiện bằng việc tổ chức hội chẩn của các bác sĩ. Điều đó chứng minh việc phát hiện kịp thời ra bệnh và mức độ mà bệnh nhân mắc phải là rất cần thiết. Hiện nay, công nghệ trí tuệ nhân tạo được biết đến với những nghiên cứu vượt bậc giúp tự động hóa những hành vi thông minh như con người, chính vì vậy, việc sử dụng công nghệ hiện đại này để phát hiện ra các mức độ của bệnh võng mạc đái tháo đường nhằm hỗ trợ các bác sĩ đưa ra những chẩn đoán chính xác. Công cụ hỗ trợ này nếu thành công, có thể giúp rút gọn thời gian chẩn đoán để các bác sĩ kịp thời đưa ra các phương pháp điều trị phù hợp.

Chúng tôi đã hình thành nên các ý tưởng và lựa chọn đề tài này làm khóa luận tốt nghiệp xuất phát từ sự hứng thú với lĩnh vực trí tuệ nhân tạo.

Ngành trí tuệ nhân tạo hiện nay đang phát triển mạnh mẽ và có nhiều công trình nghiên cứu liên quan đến y tế nói chung và biến chứng tiểu đường liên quan đến mắt nói riêng, tuy nhiên, chưa có nhiều nghiên cứu liên quan đến phát hiện bệnh võng mạc đái tháo đường. Đây chính là động lực thôi thúc chúng tôi tìm hiểu và nghiên cứu các phương pháp, công nghệ liên quan tới bài toán này.

2. Mục tiêu nghiên cứu

Mục tiêu cụ thể của khóa luận là nghiên cứu, tìm ra phương pháp xử lý dữ liệu, cách thức xây dựng mô hình mạng CNN nhằm mục dích phát hiện ra bệnh võng mạc đái tháo đường. Các mục tiêu cụ thể bao gồm:

- Thu thập dữ liệu các mức độ của bệnh dạng hình ảnh
- Xử lý dữ liệu ban đầu để có thể lựa chọn mô hình huấn luyện
- Tìm hiểu phương pháp cải thiện mô hình mạng CNN
- Đánh giá, phân tích, so sánh các mô hình đã xây dựng

3. Phương pháp nghiên cứu

Khoá luận sử dụng 2 phương pháp nghiên cứu sau:

- ***Phương pháp tổng hợp tài liệu***
- ***Phương pháp thực nghiệm***
    - Sử dụng phương pháp thu thập và tiền xử lý dữ liệu để tạo ra bộ dữ liệu huấn luyện cho mô hình
    - Áp dụng các mô hình pre-train và các phương pháp tinh chỉnh mô hình để mô hình đưa ra dự đoán tốt, độ tin cậy cao.

4. Đối tượng nghiên cứu

- **Bài toán nghiên cứu:** Tìm hiểu các phương pháp xử lý dữ liệu và các mô hình mạng CNN áp dụng cho bài toán phát hiện bệnh võng mạc đái tháo đường.
- **Dữ liệu sử dụng:**
    - Ảnh chụp võng mạc thuộc trang kaggle.com

5. Phạm vi nghiên cứu

- **Thời gian nghiên cứu:** 6 tháng
- **Dữ liệu:** Ảnh chụp đáy mắt
- **Giới hạn mô hình mạng CNN sử dụng:** VGG-16, VGG-19, Resnet-50, Resnet-101

6. Bố cục khóa luận

Mở đầu:

- Mục đích, động lực và lý do chọn đề tài khóa luận

Chương 1: Cơ sở lý thuyết

- Đầu tiên, Chương 1 giới thiệu về học máy, các mô hình CNN phổ biến trong các bài toán phân loại
- Tiếp theo, chương 1 nêu ra những vấn đề thường gặp trong quá trình huấn luyện mô hình và phương pháp xử lý

Chương 2: Thu thập và xử lý dữ liệu

- Chương 2 cung cấp thông tin về bệnh võng mạc đái tháo đường, nguồn thu thập dữ liệu, kiểu dữ liệu và cách chia dữ liệu để đào tạo mô hình
- Đưa ra các phương pháp xử lý dữ liệu như mất cân bằng, kích thức không đồng bộ,……

Chương 3: Xây dựng và tinh chỉnh mô hình

- Chương 3 của khóa luận đưa ra những mô hình áp dụng cho bài toán, các phương pháp tinh chỉnh mô hình để mô hình đạt độ chính xác tốt.

Chương 4: Mô phỏng và đánh giá

- Chương 4 mô phỏng kết quả dự đoán của mô hình và đánh giá chất lượng, độ tin cậy khi thực nghiệm.

Kết luận

- Tóm lược kết quả và đề xuất hướng phát triển cho tương lai

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

Chương này sẽ cung cấp những thông tin cơ bản về học máy, CNN, những vấn đề thường gặp trong học máy,….

1.1 Học máy

Học máy là lĩnh vực trong khoa học máy tính xây dựng các thuật toán, phương pháp để máy tính có thể học hỏi và thực hiện một nhiệm vụ nào đó dựa trên dữ liệu có sẵn.

Một cách hình thức, "học máy được định nghĩa là quá trình cải thiện hiệu suất thực hiện nhiệm vụ T (Task) xét theo độ đo P (Performance) nhờ vào kinh nghiệm E (Experience)"

(Theo G.S Tom Mitchell – Carnegie Mellon University [3])

1.2 Học sâu

Học sâu (tiếng Anh: deep learning, còn gọi là học cấu trúc sâu) là một phần trong một nhánh rộng hơn các phương pháp học máy dựa trên mạng thần kinh nhân tạo kết hợp với việc học biểu diễn đặc trưng (representation learning). Việc học này có thể có giám sát, nửa giám sát hoặc không giám sát.

(Theo G.S LeCun, Yann; Bengio, Yoshua; Hinton, Geoffrey [4])

1.3 Convolutional Neural Network (CNN) [5]

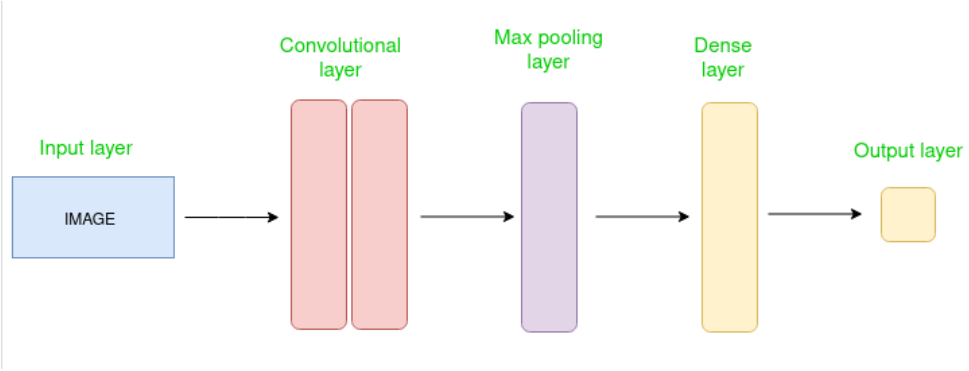
Mạng thần kinh chuyển đổi (CNN) là một loại kiến trúc thần kinh Deep Learning thường được sử dụng trong máy tính Thị giác. Thị giác máy tính là một lĩnh vực trí tuệ nhân tạo cho phép máy tính hiểu và giải thích hình ảnh hoặc dữ liệu hình ảnh.

Trong Mạng lưới thần kinh thông thường có ba loại lớp:

- Lớp đầu vào: Đây là lớp mà người dùng cung cấp đầu vào cho mô hình của mình. Số lượng tế bào thần kinh ở lớp này sử dụng tổng số tính năng trong dữ liệu (số pixel trong trường hợp là hình ảnh).
- Lớp ẩn: Đầu vào từ lớp Đầu vào sau đó được đưa vào lớp ẩn. Có thể có nhiều lớp tùy chọn ẩn trong mô hình và dữ liệu kích thước của chúng. Mỗi lớp ẩn có thể có số lượng nơ-ron khác nhau, thường lớn hơn số lượng đối tượng. Đầu ra

của mỗi lớp được tính bằng cách nhân ma trận đầu ra của lớp trước với các số có thể học được của lớp đó và sau đó bằng cách cộng các độ lệch có thể học được, sau đó là chức năng hoạt động cho mạng phi tuyến.

- Lớp đầu ra: Đầu ra từ lớp ẩn sau đó được đưa vào một hàm logistic như sigmoid hoặc softmax để chuyển đổi đầu ra của mỗi lớp thành điểm xác thực của mỗi lớp.

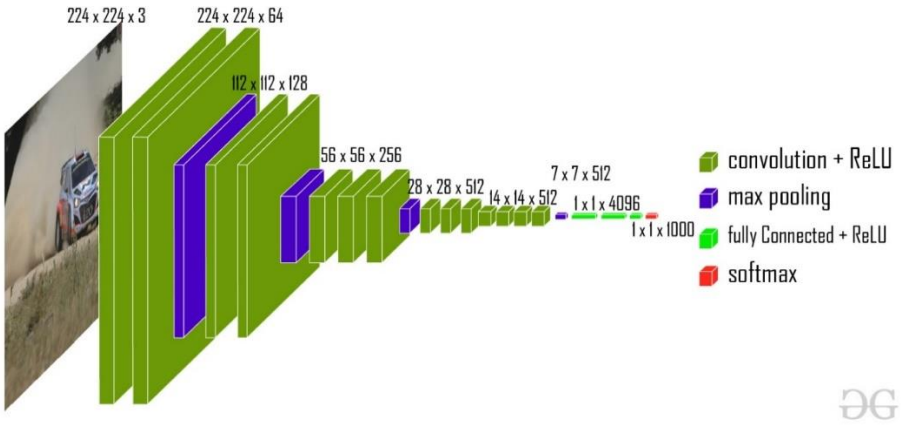


Hình 1. 1. Kiến trúc CNN đơn giản [5]

1.4 Một số mạng CNN phổ biến

1.4.1. VGG-16

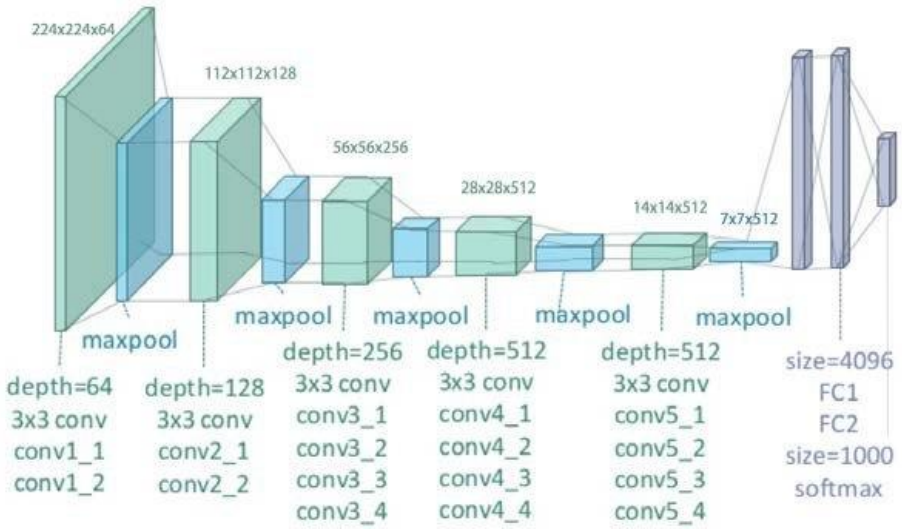
Mô hình VGG-16 là kiến trúc mạng thần kinh chớp (CNN) được sản xuất bởi Nhóm Hình học Trực quan (VGG) tại Đại học Oxford. Nó được biểu thị theo độ sâu cụ thể, bao gồm 16 lớp, trong đó có 13 lớp chậm và 3 lớp được kết nối đầy đủ. VGG-16 nổi tiếng vì tính đơn giản và hiệu quả cũng như khả năng đạt được hiệu suất mạnh mẽ trong các tác vụ thị giác máy tính khác nhau, bao gồm các phân loại hình ảnh và nhận dạng đối tượng. Cấu trúc của mô hình có nhiều lớp tích phân, sau đó là mức tối đa tổng hợp tối đa của lớp, với độ sâu tăng dần. Thiết kế này cho phép mô hình tìm hiểu cách biểu diễn phức tạp cấp độ phân cấp của các tính năng trực quan, dẫn đến những dự đoán mạnh mẽ và chính xác. Mặc dù dù đơn giản so với các kiến trúc gần đây hơn, VGG-16 vẫn là lựa chọn phổ biến cho nhiều ứng dụng deep learning do tính linh hoạt và hiệu suất cao của nó. [6]



Hình 1. 2. Kiến trúc VGG-16 [6]

1.4.2. VGG-19

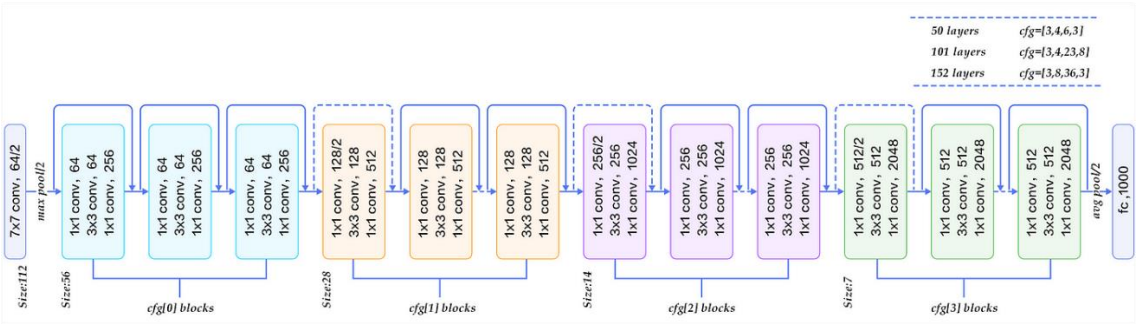
Mô hình VGG là một mạng thần kinh tích chập (CNN) có độ sâu 19 lớp. Nó được xây dựng và đào tạo bởi Karen Simonyan và Andrew Zisserman tại Đại học Oxford vào năm 2014. Mọi người có thể truy cập tất cả thông tin từ bài báo của họ, Very Deep Convolutional Networks for Large-Scale Image Recognition, được xuất bản vào năm 2015. VGG-19 được đào tạo bằng cách sử dụng hơn 1 triệu hình ảnh từ cơ sở dữ liệu ImageNet, trong đó có các ảnh màu 224×224 pixel. Đương nhiên, mọi người có thể nhập vào mô hình các trọng số được huấn luyện bởi ImageNet. Điểm đặc biệt, VGG-19 có thể phân loại tới 1000 đối tượng. [6]



Hình 1. 3. Kiến trúc VGG-19 [7]

1.4.3. Resnet-50

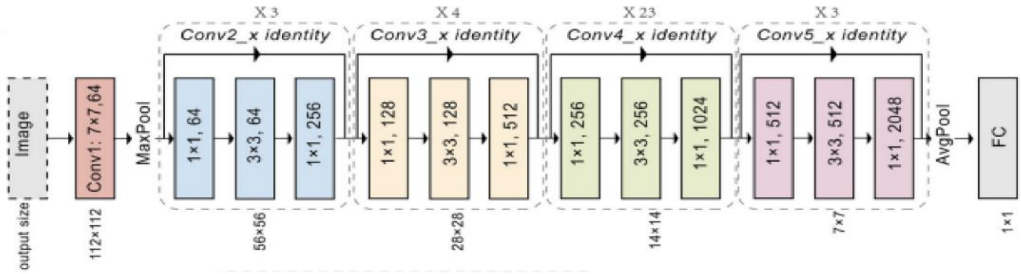
ResNet-50 là mạng nơ-ron tích chập có độ sâu 50 lớp. Bạn có thể tải phiên bản đã được huấn luyện trước của mạng nơ-ron được huấn luyện trên hơn một triệu hình ảnh từ cơ sở dữ liệu ImageNet . Mạng lưới thần kinh được huấn luyện trước có thể phân loại hình ảnh thành 1000 loại đối tượng, chẳng hạn như bàn phím, chuột, bút chì và nhiều loại động vật. Kết quả là mạng lưới thần kinh đã học được cách biểu diễn tính năng phong phú cho nhiều loại hình ảnh. Mạng thần kinh có kích thước đầu vào hình ảnh là 224 x 224. [8]



Hình 1. 4. Kiến trúc Resnet-50 [9]

1.4.4. Resnet-101

ResNet-101 là mạng nơ-ron tích chập có độ sâu 101 lớp. Bạn có thể tải phiên bản đã được huấn luyện trước của mạng đã được huấn luyện trên hơn một triệu hình ảnh từ cơ sở dữ liệu ImageNet. Mạng được huấn luyện trước có thể phân loại hình ảnh thành 1000 loại đối tượng, chẳng hạn như bàn phím, chuột, bút chì và nhiều loại động vật. Kết quả là mạng đã học được cách biểu diễn tính năng phong phú cho nhiều loại hình ảnh. Mạng có kích thước đầu vào hình ảnh là 224 x 224. [10]



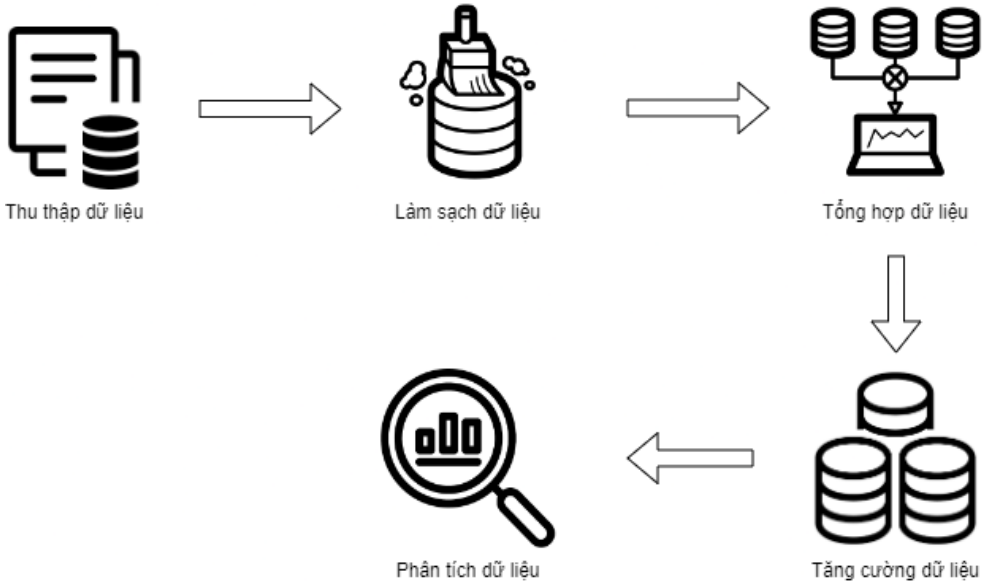
Hình 1. 5. Kiến trúc Resnet-101 [11]

1.5 Xử lý dữ liệu

Để chuẩn bị dữ liệu cho quy trình huấn luyện, ta cần tiến hành xử lý dữ liệu.

Hoạt động xử lý dữ liệu gồm các công đoạn sau

- ***Thu thập dữ liệu***: Là bước đầu tiên của công việc xử lý dữ liệu. Dữ liệu sẽ được tìm kiếm từ nhiều nguồn khác nhau và lưu trữ lại. Nguồn dữ liệu cần đáng tin cậy và chất lượng thông tin dữ liệu cần đảm bảo có tính hiệu quả cao khi sử dụng



Hình 1. 6. Các bước xử lý dữ liệu

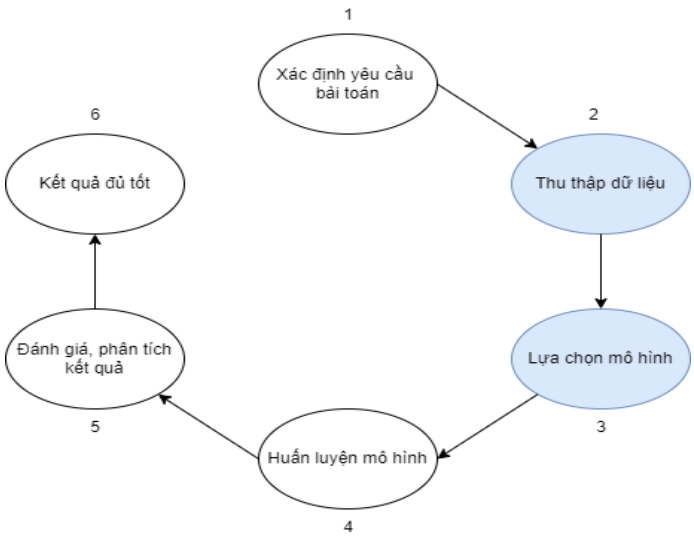
- ***Làm sạch dữ liệu***: Là quá trình sửa hoặc xóa những dữ liệu không chính xác, bị hỏng, định dạng không chính xác, trùng lặp hoặc không đầy đủ trong tập dữ liệu
- ***Tổng hợp dữ liệu***: Là bước kết hợp các tập dữ liệu thu thập được tạo thành một tập chung, thường dưới dạng bảng, nhằm mục vụ cho nhu cầu xử lý dữ liệu, phân tích và thống kê.
- ***Tăng cường dữ liệu***: Là bước áp dụng nhiều kỹ thuật khác nhau để làm tăng lượng dữ liệu hiện tại, thường bằng cách tạo ra dữ liệu mới từ dữ liệu đã có.
- ***Phân tích dữ liệu***: Là bước áp dụng một vài phương pháp thống kê mô tả, biểu đồ thống kê để phân tích nhằm hiểu hơn về dữ liệu đang khai thác, từ đó lựa chọn các phương pháp biến đổi dữ liệu phù hợp nhằm tăng tính hiệu quả.

1.6 Xây dựng mô hình học máy

Trung tâm của các bài toán học máy là quá trình ***tìm ra quy luật từ dữ liệu***. Ý tưởng chính là áp đặt một mô hình giả định lên dữ liệu và sau đó dùng máy tính tinh chỉnh các tham số của mô hình sao cho kết quả sinh ra từ đây khớp vừa kết quả thực tế. Quá trình tối ưu hoá tham số dựa trên so sánh kết quả giả định và kết quả thực tế được gọi là "huấn luyện" (*training*).

Hoạt động huấn luyện do đó sẽ tìm ra mô hình tốt nhất cho một cách huấn luyện xác định. Để thu được mô hình tối ưu, ta còn cần tìm kiếm một cách thức huấn luyện phù hợp, ví dụ như lựa chọn cách tính toán mất mát của mô hình và nhiều yếu tố khác. Các yếu tố này tạo thành các siêu tham số (*hyperparameter*) của việc huấn luyện. Việc thử nghiệm thay đổi các siêu tham số để cải thiện huấn luyện gọi là hoạt động hiệu chỉnh.

Nếu kết hợp hiệu chỉnh và huấn luyện giúp thu được mô hình đủ sát với thực tế, đây sẽ là mô hình kết quả. Nếu không, ta cần quay về hoặc hiệu chỉnh siêu tham số hoặc thay đổi bổ sung thêm dữ liệu, và cuối cùng có lúc cần phải thay đổi mô hình giả định.



Hình 1. 7. Vòng đời xây dựng một mô hình học máy

- **Bước 1**: Xác định bài toán cần giải quyết
- **Bước 2**: Thu thập dữ liệu có thể chứa kinh nghiệm giải quyết bài toán trên

- **Bước 3**: Lựa chọn một mô hình giả định mô tả dữ liệu trên
- **Bước 4**: Huấn luyện trên dữ liệu thu thập để tinh chỉnh các tham số của mô hình
- **Bước 5**: Đánh giá, phân tích kết quả của mô hình. Nếu kết quả đủ tốt thì sang bước 6, nếu không thì hoặc quay lại bước 4 để hiệu chỉnh các siêu tham số rồi huấn luyện lại; hoặc quay lại bước 3 xây dựng lại mô hình mới; hoặc quay lại bước 2 thu thập thêm dữ liệu
- **Bước 6**: Kết quả đã đủ tốt, kết thúc xây dựng mô hình.

Ngoài các bước chính trên, trên thực tế việc xây dựng một mô hình học máy sẽ còn nhiều công đoạn con như tiền xử lý dữ liệu, đánh giá thống kê tương quan dữ liệu,v.v.

1.7 Siêu tham số

Trong quy trình thực hiện dự án trong phần 1.4 trên, tại bước 5 Đánh giá, phân tích kết quả của mô hình, nếu đã thu được kết quả đủ tốt thì ta chuyển sang bước 6 kết thúc huấn luyện mô hình, nếu không thì hoặc quay lại bước 4 hiệu chỉnh các siêu tham số rồi huấn luyện lại mô hình, hoặc quay lại bước 3 xây dựng lại mô hình mới, hoặc quay lại bước 2 thu thập thêm dữ liệu.

Dưới đây là một vài siêu tham số thường được sử dụng:

- ***Learning rate***: Tốc độ học là một siêu tham số sử dụng trong quá trình huấn luyện mô hình. Giá trị là một số dương, nằm trong khoảng 0 và 1. Tốc độ học giúp kiểm soát tốc độ thay đổi các trọng số của mô hình trong khi tính toán *gradient descent.*
- ***Batch size***: Là kích thước của mẫu trong một lần huấn luyện
- ***Epoch***: Là một lần mô hình học toàn bộ tập huấn luyện

1.8 Các vấn đề cần lưu ý trong học máy

1.8.1. Underfitting và Overfitting

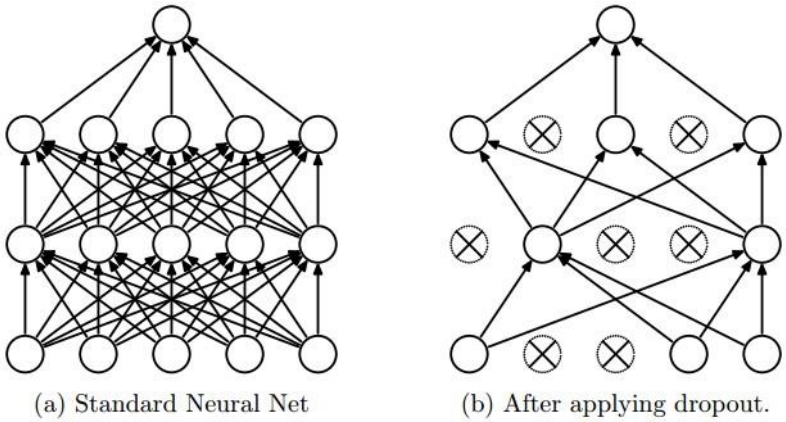
Khi xây dựng và huấn luyện một mô hình thường xảy ra một số vấn đề không mong muốn như mô hình dự đoán không chính xác, mô hình chỉ chính xác trên tập huấn luyện hay sai số quá lớn, ... Để tránh các trường hợp trên xảy ra, khi xây dựng

mô hình, chúng ta cần nắm được một số khái niệm cũng như kỹ thuật để giải quyết các vấn đề này. Dưới đây mô tả 2 vấn đề thường gặp trong quá trình huấn luyện:

- ***Underfitting***: Là việc lựa chọn mô hình chưa phù hợp với tập dữ liệu huấn luyện và các mẫu mới khi dự đoán. Nguyên do vì mô hình chưa đủ độ phức tạp cần thiết để bao quát được tập dữ liệu.
- ***Overfitting***: Là hiện tượng mô hình quá khớp với tập dữ liệu huấn luyện, điều này sẽ gây ra hậu quả nghiêm trọng nếu tập dữ liệu huấn luyện xuất hiện nhiễu. Mô hình sẽ chỉ chú trọng xấp xỉ với tập dữ liệu huấn luyện mà không đạt đến tổng quát hóa, kết quả là mô hình sẽ không có khả năng dự đoán thật sự tốt đối với dữ liệu nằm ngoài tập huấn luyện (ví dụ như dữ liệu kiểm thử và dữ liệu thực tế). Overfitting nói chung xảy ra khi độ phức tạp của mô hình quá lớn hoặc quá ít dữ liệu.

1.8.2. Phương pháp xử lý các vấn đề trong học máy

- **Drop out:** Nguyên lý chính của phương pháp drop-out là tạm tắt bỏ một số đơn vị trong các bước huấn luyện, từ đó giúp quá trình học nhanh hơn đồng thời có khả năng giúp tránh overfitting tốt hơn.



Hình 1. 8. Drop-out [12]

Cách hoạt động của drop-out:

- Với mỗi lớp của mạng neuron, dropout được áp dụng với một xác suất *p* cho trước (có thể sử dụng nhiều drop-out khác nhau cho những layer khác nhau, nhưng trên 1 layer sẽ chỉ có 1 dropout).

- Tại mỗi bước trong quá trình huấn luyện, khi thực hiện truyền thẳng đến lớp sử dụng drop-out, thay vì tính toán tất cả đơn vị ở trên lớp thì tại mỗi đơn vị, ta sẽ thực hiện "gieo xúc xắc" với xác suất *p* xem đơn vị đó được sử dụng (*active*) hay không được sử dụng (*deactive*). Những đơn vị sử dụng sẽ được tính toán bình thường còn những đơn vị không được sử dụng sẽ được đặt giá trị bằng 0.
- Trong quá trình kiểm thử, tất cả các đơn vị đều được sử dụng và đầu ra mong đợi của các đơn vị sẽ giống với đầu ra mong đợi của quá trình huấn luyện.

- ***Tăng cường dữ liệu (Data augmentation):*** Bằng cách tạo thêm dữ liệu từ dữ liệu có sẵn, tăng cường dữ liệu là một phương pháp làm đa dạng dữ liệu từ đó giúp máy học được nhiều loại kinh nghiệm hơn, từ đó làm tăng tính tổng quát và giúp hạn chế Overfitting. Dưới đây là 2 kịch bản chủ yếu cần tăng cường dữ liệu
  - Ta sở hữu một tập dữ liệu quá nhỏ, tập này có thể khiến mô hình không học được các đặc trưng quan trọng do dữ liệu đầu vào không đủ. Do đó, chúng ta cần làm tăng dữ liệu.
  - Khi ta có lượng lớn dữ liệu, nhưng tập dữ liệu đó lại mất cân bằng, có nghĩa là tồn tại chênh lệch số lượng lớn giữa các lớp của tập dữ liệu. Khi đó, mô hình có xu hướng sẽ học kĩ đặc trưng của những lớp có số lượng phần tử nhiều hơn, khiến mô hình trở thành học tủ, và những lớp ít dữ liệu bị học kém hơn so với các lớp có nhiều dữ liệu, mô hình do đó sẽ không đạt hiệu quả tốt. Nếu có thể tăng số lượng các lớp ít dữ liệu để đạt đến sự cân bằng, mô hình sẽ có khả năng học được tốt hơn.
  - Với những trường hợp trên, chúng ta đều có thể thử nghiệm tăng cường dữ liệu bằng một số kỹ thuật như sau:
  - Đối với bài toán xử lý ảnh: Dãn kích thước ảnh, dịch chuyển ảnh, tăng giảm độ sáng, thêm độ nhiễu cho ảnh, …

- Đối với bài toán xử lý ngôn ngữ tự nhiên: Đổi chỗ các một vài cặp từ trong câu, xóa một từ bất kỳ trong câu, tìm thay thế một một từ trong câu với một từ đồng nghĩa tương ứng, …

- ***Early stopping***: Khi dùng một phương pháp tối ưu hàm số để giảm thiểu giá trị mất mát thì J_{train} , J_{val} sẽ cùng giảm theo thời gian nhưng nếu sau một thời gian J_{val} tăng lên còn J_{train} tiếp tục giảm thì đó là lúc bắt đầu dẫn đến overfitting. Cách đơn giản nhất để giảm thiểu overfitting là dừng huấn luyện tại ngay thời điểm bắt đầu overfitting, phương pháp này được gọi là early stopping. Phương pháp này hướng tới nhận biết khi nào mô hình có dấu hiệu overfitting từ đó có thể xây dựng chiến thuật để xử lý. Việc phát hiện overfitting sẽ giảm thiểu thời gian và công sức lãng phí.

- **Regularization**: là kỹ thuật để giải quyết một bài toán giả định sai hoặc để ngăn chặn sự quá khớp (overfitting). Có thể hiểu Regularization là bất kỳ thay đổi nào mà chúng ta tạo ra với một thuật toán học thuật nhằm giảm lỗi tổng quát chứ không phải là lỗi huấn luyện.

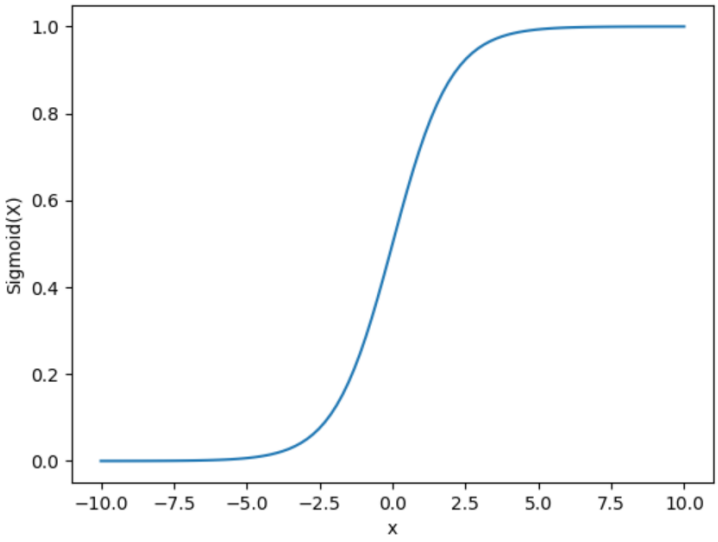
1.8.3. Hàm kích hoạt (activation) [13]

Định nghĩa: Hàm kích hoạt (activation function) là một thành phần quan trọng trong mạng nơ-ron nhân tạo. Hàm kích hoạt quyết định liệu một nơ-ron có nên được kích hoạt hay không bằng cách tính tổng trọng số và thêm độ lệch cho nó. Mục đích của hàm kích hoạt là đưa tính phi tuyến tính vào đầu ra của nơ-ron.

Một số hàm kích hoạt thường dùng:

- **Hàm sigmoid**
  - Đây là một hàm được vẽ dưới dạng đồ thị hình chữ 'S' .
  - Phương trình : A = 1/(1 + e -x)
  - Bản chất: Phi tuyến tính. Lưu ý rằng giá trị X nằm trong khoảng từ -2 đến 2, giá trị Y rất dốc. Điều này có nghĩa là những thay đổi nhỏ của x cũng sẽ dẫn đến những thay đổi lớn về giá trị của Y.
  - Phạm vi giá trị: 0 đến 1

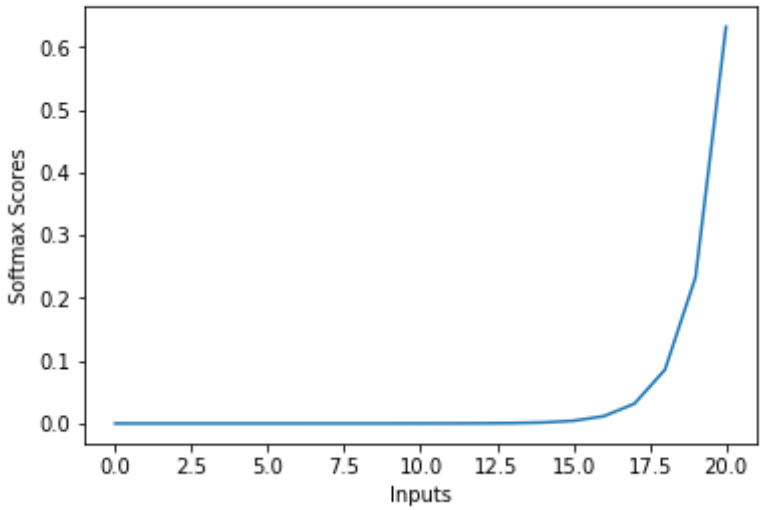
- Công dụng: Thường được sử dụng trong lớp đầu ra của phân loại nhị phân, trong đó kết quả là 0 hoặc 1, vì giá trị của hàm sigmoid chỉ nằm trong khoảng từ 0 đến 1 nên kết quả có thể dễ dàng được dự đoán là 1 nếu giá trị lớn hơn 0,5 và 0 nếu không .



Hình 1. 9. Hàm kích hoạt sigmoid [14]

- **Hàm RELU**
  - Nó là viết tắt của đơn vị tuyến tính được chỉnh lưu . Đây là chức năng kích hoạt được sử dụng rộng rãi nhất. Được triển khai chủ yếu ở các lớp ẩn của mạng nơ-ron.
  - Phương trình :- A(x) = max(0,x) . Nó cho đầu ra x nếu x dương và 0 nếu ngược lại.
  - Phạm vi giá trị :- [0, inf)
  - Bản chất: - phi tuyến tính, có nghĩa là chúng ta có thể dễ dàng truyền ngược các lỗi và có nhiều lớp nơ-ron được kích hoạt bởi hàm ReLU.
  - Công dụng: - ReLu ít tốn kém về mặt tính toán hơn tanh và sigmoid vì nó bao gồm các phép toán đơn giản hơn. Tại một thời điểm, chỉ có một số nơ-ron được kích hoạt khiến mạng trở nên thưa thớt, giúp việc tính toán trở nên hiệu quả và dễ dàng.

- **Hàm Softmax**



Hình 1. 10. Hàm kích hoạt Softmax [15]

- Hàm softmax cũng là một loại hàm sigmoid nhưng rất hữu ích khi chúng ta đang cố gắng xử lý các vấn đề phân loại nhiều lớp.
- Bản chất: - phi tuyến tính
- Công dụng: - Thường được sử dụng khi cố gắng xử lý nhiều lớp. Hàm softmax thường được tìm thấy trong lớp đầu ra của các bài toán phân loại hình ảnh. Hàm softmax sẽ nén các đầu ra của mỗi lớp trong khoảng từ 0 đến 1 và cũng sẽ chia cho tổng các đầu ra.
- Đầu ra: - Hàm softmax được sử dụng lý tưởng trong lớp đầu ra của bộ phân loại nơi chúng ta thực sự đang cố gắng đạt được xác suất để xác định lớp của từng đầu vào.
- Nguyên tắc cơ bản là nếu bạn thực sự không biết nên sử dụng chức năng kích hoạt nào thì chỉ cần sử dụng RELU vì đây là chức năng kích hoạt chung trong các lớp ẩn và được sử dụng trong hầu hết các trường hợp hiện nay.
- Nếu đầu ra của bạn dành cho phân loại nhị phân thì hàm sigmoid là sự lựa chọn rất tự nhiên cho lớp đầu ra.
- Nếu đầu ra của bạn dành cho phân loại nhiều lớp thì Softmax rất hữu ích để dự đoán xác suất của từng lớp.

1.8.4. Hàm mất mát (Loss function)

Huấn luyện mạng neuron nhân tạo cũng giống với cách dạy con người học tập. Mô hình dữ liệu được huấn luyện sẽ đưa ra dự đoán, để cải thiện huấn luyện thì cần phải có sự phản hồi xem dự đoán đó có chính xác hay không. Đó là lý do hàm tính lỗi được thiết kế. Hàm tính lỗi sẽ ước lượng mô hình đoán sai bao nhiêu so với giá trị thực tế.

Hàm mất mát cross-entropy được dùng để đo sự giống nhau của hai phân phối xác suất. Giá trị của hàm số càng nhỏ thì hai xác suất càng gần nhau. Hàm này thường được sử dụng cho các bài toán phân loại.

- p_i là giá trị đầu ra của một mẫu, phân bố xác suất $p = [p_1, p_2, \dots, p_n]$ với $p_i \epsilon [0, 1]$
- q_i là giá trị đầu ra của một mẫu, phân bố xác suất $q = [q_1, q_2, \dots, q_n]$ với $q_i \neq 0 \; \forall i$

Công thức tính cross entropy giữa hai phân bố p và q là:

$$H(p, q) = -\sum_{i=1}^{n} p_i \log q_i$$

Trong đó n là số lượng các class cần phân lớp

1.8.5. Phương pháp tối ưu (optimizer)

Mục tiêu của các phương pháp tối ưu là tìm ra nghiệm tại đó hàm chi phí đạt giá trị nhỏ nhất. Tuy nhiên, việc tìm điểm cực tiểu toàn cục (global minimum) rất phức tạp, có lúc bất khả thi. Để giải quyết vấn đề này, chúng ta thường cố gắng tìm các điểm lân cận gần nhất ở một mức độ nào đó và coi điểm đó là nghiệm của bài toán.

Một số thuật toán tối ưu hiện nay là: gradient descent, stochastic gradient descent, adam, rmsprop,…

Dưới đây mô tả chỉ tiết hơn về phương pháp tối ưu sử dụng trong khoá luận này:

- ***Adaptive moment estimation (Adam)*** [16]

Adam optimizer là một thuật toán kết hợp kỹ thuật của RMS prop và momentum. Thuật toán sử dụng hai internal states momentum (m) và squared momentum (v) của gradient cho các tham số. Sau mỗi lô dữ liệu huấn luyện, giá trị của *m* và *v* được cập nhật lại sử dụng trung bình trượt số mũ (exponential weighted averaging) như sau:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \ (1)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \ (2)$$

trong đó $\beta_1$và β_2 là hai siêu tham số thường được chọn: $\beta_1 = 0.9$ và $\beta_2 = 0.999$.

Công thức cập nhật θ như sau:

$$\theta_t = \theta_{t-1} - \alpha \frac{m^t}{\sqrt{v_t} + \in}$$

trong đó α là tốc độ học (*learning rate*), $\in$ là giá trị được thêm vào để tránh việc chia cho 0.

Để giúp cho việc trượt dốc (*descent*) được thực hiện nhanh hơn, thuật toán đã sử dụng hai kỹ thuật:

- Tính trung bình trượt số mũ của giá trị đạo hàm lưu vào biến *m* và dùng nó là tử số trong công thức cập nhật hướng. Nếu *m* có giá trị lớn, việc trượt dốc đang đi đúng hướng và cần bước nhảy lớn hơn để đi nhanh hơn. Nếu giá trị *m* nhỏ, phần trượt dốc có thể không đi về hướng tối thiểu, do đó ta chỉ nên đi những bước nhỏ để tiếp tục thăm dò. Đây là ý nghĩa khía cạnh momentum của thuật toán.
- Tính trung bình trượt số mũ của bình phương giá trị đạo hàm lưu vào biến *v* và sử dụng nó là phần mẫu số của việc cập nhật hướng, với ý nghĩa như sau: Giả sử gradient mang các giá trị âm, dương lẫn lộn, khi cộng các giá trị lại

theo công thức tính m, ta sẽ thu được giá trị *m* gần với 0. Do âm, dương lẫn lộn nên các giá trị này triệt tiêu lẫn nhau. Tuy nhiên trong trường hợp này, *v* lại mang giá trị lớn. Do đó, quá trình huấn luyện đang không hướng về cực tiểu, và chúng ta sẽ không muốn đi theo hướng của đạo hàm trong trường hợp này. Do đó, ta sẽ để *v* ở phần mẫu vì một giá trị *v* cao sẽ kéo theo giá trị của các phần cập nhật sẽ nhỏ, và ngược lại, khi *v* có giá trị nhỏ thì giá trị phần cập nhật sẽ lớn. Đây chính là ý nghĩa của khía cạnh RMSProp.

Ở đây, *m* được xem như là moment thứ nhất, *v* xem như là moment thứ hai, nên thuật toán còn có tên là "Adaptive moment estimation".

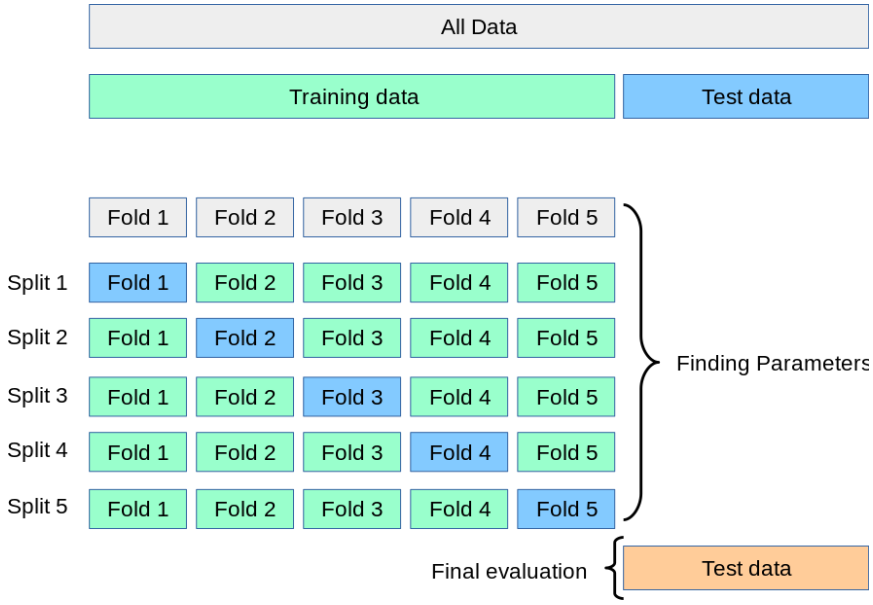
1.8.6. Cross validation

Khi huấn luyện mô hình ta thường chia thành tập huấn luyện và tập kiểm thử. Việc mô hình học và kiểm tra lặp lại các mẫu mà nó vừa thấy sẽ dễ gây ra tình trạng học quá khớp (overfitting). Để giải quyết vấn đề này, một phần tập huấn luyện được lấy làm tập xác thực. Quá trình đào tạo tiến hành trên tập huấn luyện và đánh giá trên tập xác thực, khi kết quả có vẻ khả quan thì đánh giá cuối cùng trên tập kiểm thử.

Tuy nhiên, nếu chia thành ba tập sẽ làm giảm đáng kể lượng mẫu để mô hình học và kết quả có thể phụ thuộc vào lượng mẫu ngẫu nhiên của tập huấn luyện và tập xác thực. Vậy nên cross validation (CV) là giải pháp cho vấn đề nêu trên. Trong cách tiếp cận cơ bản, được gọi là K-Fold CV, tập train được chia thành k tập nhỏ hơn. Quy trình sau đây được thực hiện cho từng Fold.

- Một mô hình được huấn luyện bằng cách sử dụng k-1 fold làm dữ liệu huấn luyện
- Mô hình kết quả được xác thực trên phần còn lại của dữ liệu.

Thước đo hiệu suất được báo cáo bởi xác thực chéo k -fold khi đó là giá trị trung bình của các giá trị được tính toán trong vòng lặp. Cách tiếp cận này có thể tốn kém về mặt tính toán nhưng không lãng phí quá nhiều dữ liệu (như trường hợp sửa một bộ xác thực tùy ý), đây là một lợi thế lớn trong các vấn đề như suy luận nghịch đảo trong đó số lượng mẫu rất nhỏ.



Hình 1. 11. Cách hoạt động của k-fold CV [17]

1.9 Các hình thức đánh giá mô hình

Để đảm bảo chất lượng một mô hình học máy, chúng ta cần có các phương pháp đánh giá hay là các phép đo tính hiệu quả. Các phép đo này thường tính toán trên các vector dự đoán đầu ra của tập kiểm thử. Mỗi vector dự đoán được so sánh với vector nhãn thật của dữ liệu từ đó cho ra một ước lượng sai số, tổng sai số có thể cho ta biết hiệu quả của mô hình.

Giả sử rằng:

- ***Positive*** (P): Tin tức tốt
- ***Negative*** (N): Tin tức xấu
- ***True positive*** (TP): Là số lượng tin tức dự đoán đúng là tốt và thực tế tin tức tốt
- ***False positive*** (FP): Là số lượng tin tức dự đoán sai là tốt nhưng thực tế tin tức xấu
- ***True negative*** (TN): Là số lượng tin tức dự đoán đúng là xấu và thực tế tin tức xấu
- ***False negative*** (FN): Là số lượng tin tức dự đoán sai là xấu nhưng thực tế tin tức tốt

Khoá luận sẽ sử dụng độ đo độ chính xác (accuracy) để đánh giá mô hình xây dựng với công thức sau:

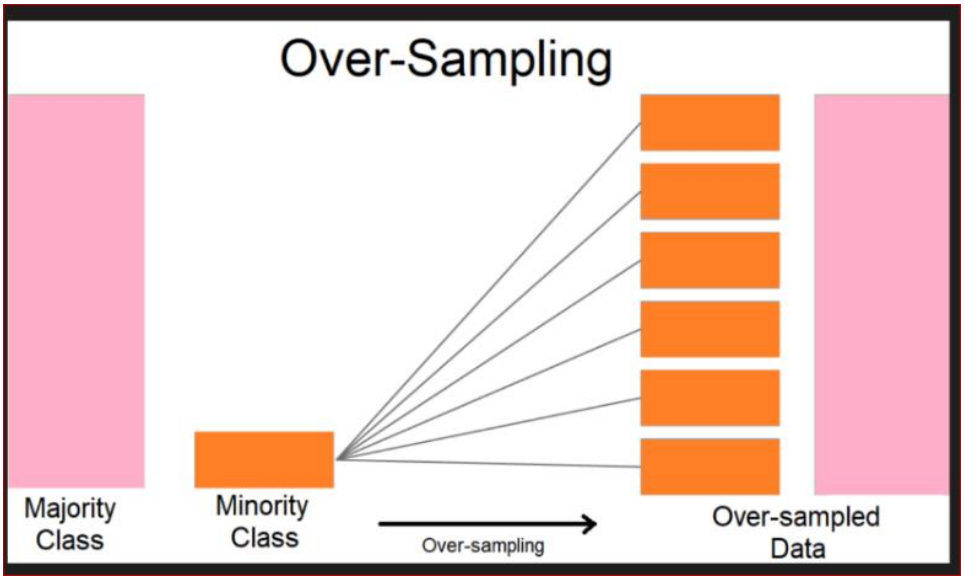
$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

1.10 Mất cân bằng dữ liệu

Trong học máy việc mất cân bằng dữ liệu là thường gặp. Tuy nhiên việc các lớp có sự chêch lệch quá cao thì mô hình không tổng quát được tất cả các lớp. Mô hình chỉ tập trung vào những lớp có nhiều dữ liệu nên dự đoán tốt còn những lớp ít dữ liệu thì dự đoán kém. Vậy nên việc xử lý mất cân bằng rất quan trọng. Dưới đây là một số các xử lý mất cân bằng dữ liệu:

1.9.1. OverSampler

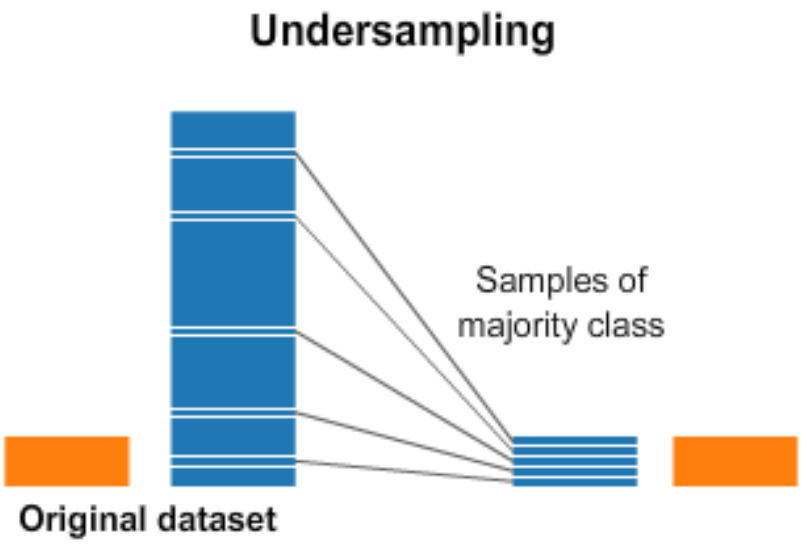
OverSampler là lấy mẫu quá mức. Những lớp có ít dữ liệu sẽ được sao chép vào nhân bản lên để có số lượng xấp xỉ bằng với những lớp có nhiều dữ liệu hơn. Tuy nhiên phương pháp này do phải nhân bản nên dễ dẫn đến overfitting. *Hình 1.12* dưới đây mô phỏng rõ hơn phương pháp OverSampler.



Hình 1. 12. Phương pháp over sampling [18]

1.9.2. UnderSampler

UnderSampler là lấy mẫu dưới mức. Những lớp có nhiều dữ liệu sẽ bị cắt bỏ bớt để có số lượng xấp xỉ bằng số lượng của những lớp ít dữ liệu hơn. Tuy nhiên việc này gây ra mất dữ liệu nếu dữ liệu quá ít mô hình có thể bị underfitting. *Hình 1.13* dưới đây mô phỏng rõ hơn phương pháp UnderSampler.



Hình 1. 13. Phương pháp under sampling [19]

1.9.3. Đánh trọng số lớp

Sự chênh lệch lớn giữa các lớp khiến cho mô hình lơ là việc dự đoán những lớp có ít dữ liệu mà chỉ tập trung vào những lớp có nhiều dữ liệu. Vậy nên việc đánh trọng số sẽ khiến mô hình "chú ý nhiều hơn" đến những lớp ít dữ liệu. Công thức đánh trọng số lớp:

Weight_class = (1/class)*(total/n_class)

Trong đó:

- class: số lượng mẫu trong lớp
- total: tổng số lượng mẫu của n_class
- n_class: số lượng nhãn

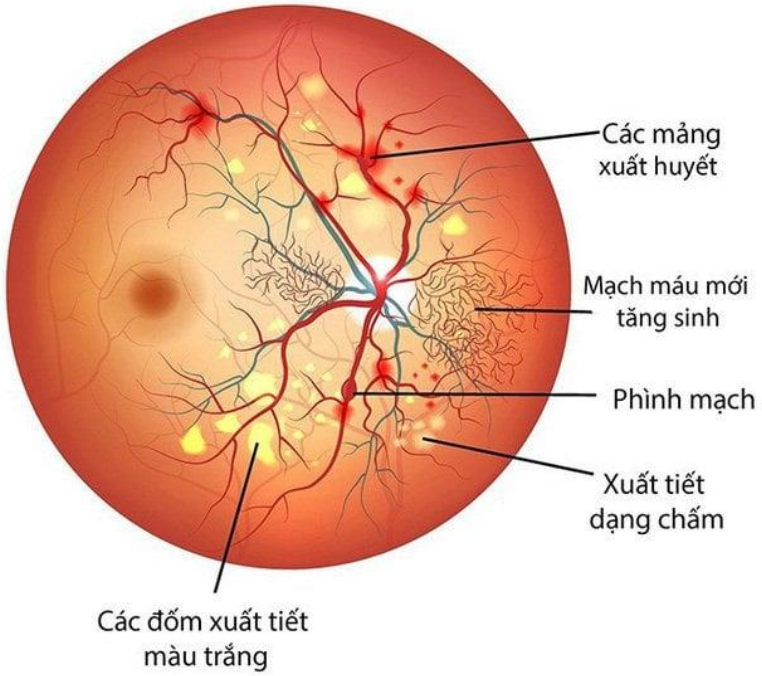
CHƯƠNG 2: THU THẬP VÀ XỬ LÝ DỮ LIỆU

Trong phần này, khóa luận sẽ mô tả về dữ liệu, nêu rõ nguồn thu thập dữ liệu đã thu thập được và sau đó sẽ thực hiện thống kê dữ liệu sử dụng các biểu đồ.

2.1 Thu thập và phân tích dữ liệu

2.1.1 Bệnh võng mạc đái tháo đường [20]

- **Nguyên nhân**: Bệnh đái tháo đường gây nên tổn thương các mạch máu của toàn bộ các cơ quan trong cơ thể, biểu hiện rõ nhất ở các vi mạch máu. Tại mắt, do tổn thương các mao mạch võng mạc, làm tăng tính thấm thành mạch, thoát huyết tương vào võng mạc gây phù nề. Khi mao mạch bị phá hủy gây tắc và làm thiếu máu võng mạc, khi đó cơ thể phản ứng bằng cách tiết ra các yếu tố kích thích sự phát triển các mạch máu mới (tân mạch) để nuôi dưỡng những vùng võng mạc này. Tuy nhiên những mạch máu này mỏng manh dễ vỡ gây ra các biến chứng xuất huyết dịch kính, xơ hóa gây co kéo bong võng mạc.
- **Các giai đoạn của bệnh**:
    - **Bệnh lý võng mạc nền (**Mild): Đây là giai đoạn sớm của bệnh lý võng mạc do đái tháo đường với những tổn thương như phình mao mạch võng mạc, xuất huyết nhẹ, ứ đọng các chất tiết trong võng mạc, phù võng mạc.
    - **Bệnh lý hoàng điểm do đái tháo đường (**Moderate): Phù hoàng điểm với các xuất tiết cứng quanh hoàng điểm
    - **Bệnh lý võng mạc đái tháo đường tiền tăng sinh (**Severe): Tổn thương ở võng mạc giai đoạn này gây nên bởi sự bất thường cung cấp máu cho võng mạc, dẫn đến các tổn thương thiếu máu cục bộ, xuất huyết, xuất tiết và phù võng mạc.
    - **Bệnh lý võng mạc đái tháo đường giai đoạn tăng sinh** (Proliferative DR): Bệnh lý giai đoạn này gây ra bởi sự tăng sinh các tân mạch bất thường, gây xuất huyết tái diễn liên tục, gây tổ chức hóa và co kéo dịch kích võng mạc.



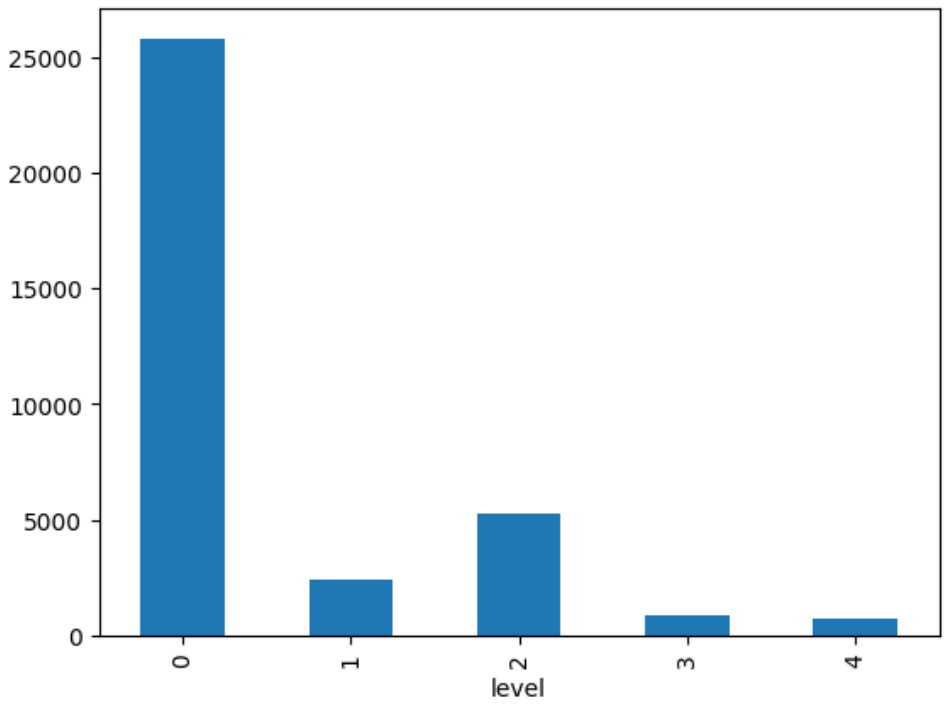
Hình 2. 1. Bệnh võng mạc đái tháo đường [21]

2.1.2. Thu thập dữ liệu

- Nguồn thu thập dữ liệu: Dữ liệu được thu thập tại trang kaggle.com
- Dữ liệu thu được gồm:
    - Thư mục bao gồm các ảnh chụp võng mạc, những ảnh này được chụp dưới nhiều điều kiện ánh sáng, kích thước,…. khác nhau
    - Một file csv lưu trữ tên ảnh và nhãn của ảnh. Ảnh được gán nhãn bởi bác sĩ và chia theo 5 mức độ của bệnh (không có bệnh và 4 giai đoạn – Mild, Moderate, Severe, Proliferative DR)

2.1.3. Phân tích dữ liệu

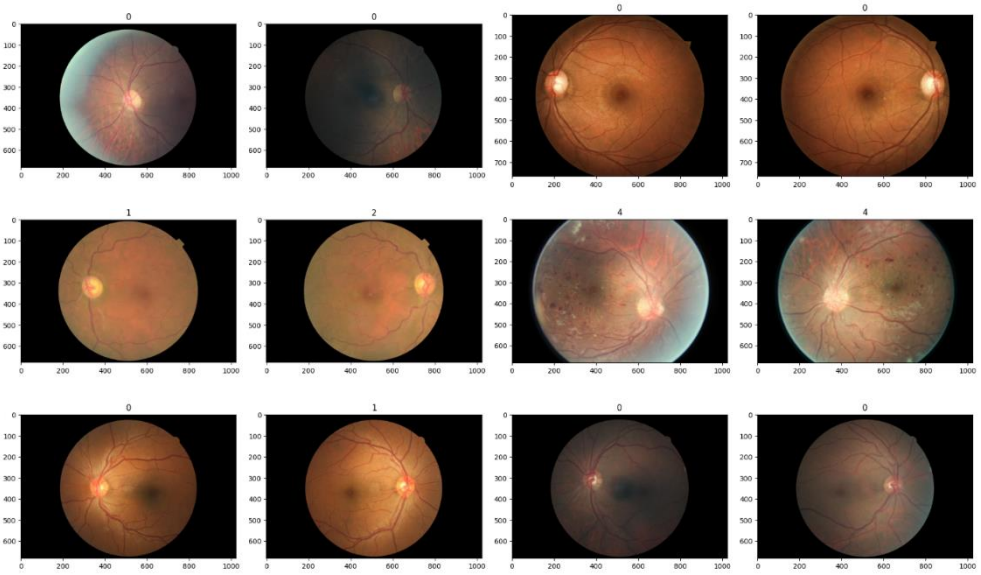
- Dữ liệu bao gồm:
    - Khoảng 35 000 bức ảnh chụp võng mạc
    - 5 lớp tương ứng với 5 cấp độ của bệnh:
        - 0 - No DR: Bình thường
        - 1 - Mild
        - 2 - Moderate
        - 3 - Severe
        - 4 - Proliferative DR: giai đoạn bệnh lý nặng nhất.



Hình 2. 2. Biểu đồ tổng quan về số lượng ảnh của từng lớp

Nhận xét:

- Nhìn vào biểu đồ ta thấy dữ liệu thu gặp vấn đề mất cân bằng dữ liệu.
- Nhãn không mắc bệnh có số lượng ảnh nhiều nhất với khoảng 25 000 bức ảnh. (Do số lượng ảnh võng mạc không mắc bệnh sẽ có rất nhiều và dễ tìm kiếm hơn)
- 10 000 bức ảnh còn lại thuộc vào 4 lớp còn lại:
    - Nhãn Mild có khoảng 2500 bức ảnh
    - Nhãn Moderate có khoảng 5000 bức ảnh.
    - Nhãn Severe có khoảng 800 bức ảnh
    - Nhãn Proliferative DR có khoảng 700 bức ảnh
- Với sự mất cân bằng này mô hình sẽ học nhiều ở những lớp nhiều, mô hình không tổng quát được tất cả các lớp và không đáng tin cậy



Hình 2. 3. Hình ảnh chụp võng mạc

Nhận xét:

- Ảnh chụp võng mạc có kích thước khác nhau và có nền màu đen không đem lại giá trị trong quá trình học
- Điều kiện ánh sáng của các ảnh khác nhau: ảnh mờ, sáng, tối,…
- Việc màu sắc, ánh sáng cũng gây ảnh hưởng đến độ chính xác trong quá trình học của máy

2.2 Tiền xử lý dữ liệu

Trong phần này, khóa luận sẽ nêu rõ các bước thực hiện xử lý dữ liệu tđã thu thập được. Như ở mục 2.1.3 đã nêu ra các vấn đề của dữ liệu ảnh thu. Vậy nên cần phải tiền xử lý dữ liệu trước khi đưa vào huấn luyện. Chi tiết sẽ được nêu ra dưới đây:

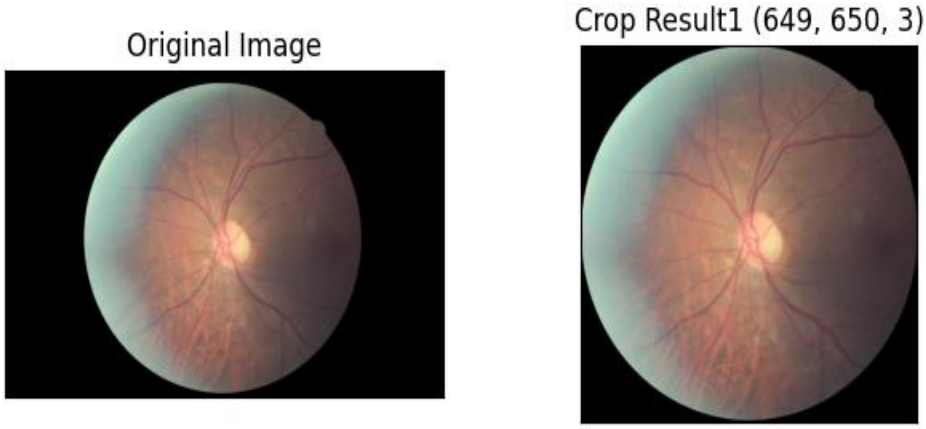
2.2.1 Crop ảnh, Điều chỉnh màu sắc, ánh sáng

- Nền đen không đóng góp được nhiều trong quá trình học của mô hình gây nhiễu. Dưới đây là cách mà khóa luận thực hiện loại bỏ nền đen cho dữ liệu:

```
def crop_image_from_gray(img, tol=7):
    if img.ndim == 2:
        mask = img > tol
        return img[np.ix_(mask.any(1), mask.any(0))]
    elif img.ndim == 3:
        gray_img = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)
        mask = gray_img > tol
```

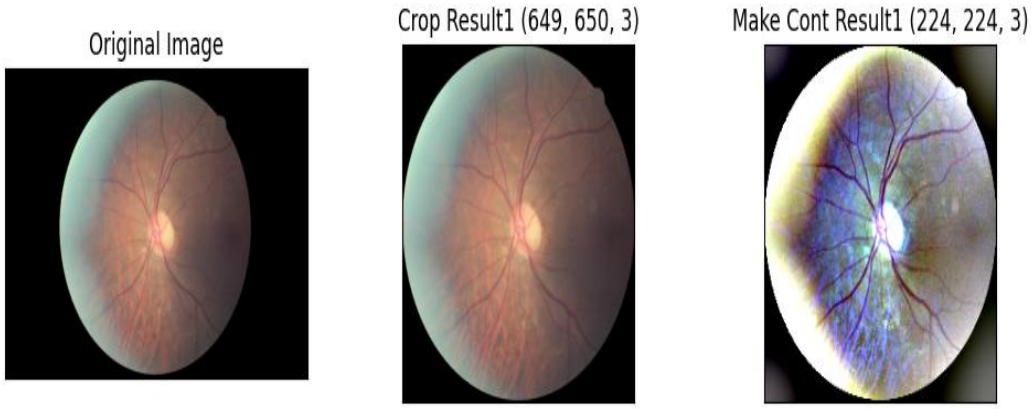
```
        check_shape = img[:, :, 0][np.ix_(mask.any(1),
mask.any(0))].shape[0]
        if check_shape == 0:
            return img
        else:
            img1 = img[:, :, 0][np.ix_(mask.any(1), mask.any(0))]
            img2 = img[:, :, 1][np.ix_(mask.any(1), mask.any(0))]
            img3 = img[:, :, 2][np.ix_(mask.any(1), mask.any(0))]
            img = np.stack([img1, img2, img3], axis=-1)
        return img
```

Hình 2.4 dưới đây là kết sau khi thực hiện cắt bỏ nền đen:



Hình 2. 4. Ảnh võng mạc sau khi loại bỏ nền đen

- Sử dụng GaussianBlur của thư viện OpenCV để tăng cường độ tương phản để làm lộ ra những đường mạch máu, điểm xuất huyết. Đồng thời đưa kích thức ảnh về 224x224 để có thể thực hiện huấn luyện



Hình 2. 5. Kết quả của điều chỉnh màu sắc, ánh sáng

Hình 2.5 là kết quả của các bước xử lý ánh sáng dưới đây:

```
def make_cont(img, to_gray=False, IMG_SIZE=224):
    """
    Increase image contract
    """
    if to_gray:
        img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    else:
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    img = cv2.resize(img, (IMG_SIZE, IMG_SIZE))
    cimg = cv2.addWeighted(img, 4, cv2.GaussianBlur(img, (0, 0),
IMG_SIZE / 10), -4, 128)
    return cimg
```

2.2.2 Cân bằng dữ liệu

- **Oversampling:** Do 4 nhãn Mild, Moderate, Severe, Proliferative DR có số lượng dữ liệu ít hơn nhiều so với nhãn No DR nên khi áp dụng phương pháp oversampling thì dữ liệu của 4 nhãn trên sẽ được sao chép và nhân bản lên. Sau khi tăng lượng dữ liệu ta có kết quả sau:
    - Nhãn No DR: 25000
    - Nhãn Mild: 25000
    - Nhãn Moderate: 25000
    - Nhãn Severe: 25000
    - Nhãn Proliferative DR: 25000
- Dưới đây là cách mà khóa luận tăng số lượng mẫu cho bộ dữ liệu

```
def balance_data(class_size,df):
    train_df = df.groupby(['level']).apply(lambda x:
x.sample(class_size, replace = True)).reset_index(drop = True)
    train_df = train_df.sample(frac=1).reset_index(drop=True)
    print('New Data Size:', train_df.shape[0], 'Old Size:',
df.shape[0])
    train_df['level'].hist(figsize = (10, 5))
    return train_df
train_data = balance_data(25000,train_df)
```

Tuy nhiên, do số lượng của các lớp chênh lệch quá lớn, nhãn No DR có lượng dữ liệu gấp 5 lần nhãn Moderate, 10 lần nhãn Mild và khoảng 15 - 20 lần với nhãn Severe, Proliferative DR . Vậy nên việc sử dụng oversampling cho bộ dữ liệu này có thể làm cho mô hình bị overfitting.

- **Undersampling:** Do số lượng dữ liệu của nhãn No DR, Mild, Moderate, Severe lớn hơn nhiều so với nhãn Proliferative DR nên khi áp dụng phương pháp undersampling thì dữ liệu của 4 nhãn trên sẽ được cắt bớt đi. Dưới đây là cắt bớt số lượng mẫu trong những lớp No DR, Mild, Moderate, Severe

```
def balance_data(class_size,df):
    train_df = df.groupby(['level']).apply(lambda x:
x.sample(class_size, replace = True)).reset_index(drop = True)
    train_df = train_df.sample(frac=1).reset_index(drop=True)
    print('New Data Size:', train_df.shape[0], 'Old Size:',
df.shape[0])
    train_df['level'].hist(figsize = (10, 5))
    return train_df
train_data = balance_data(700,train_df)
```

- Sau khi giảm lượng dữ liệu ta có kết quả sau:
    - Nhãn No DR: 700
    - Nhãn Mild: 700
    - Nhãn Moderate: 700
    - Nhãn Severe: 700
    - Nhãn Proliferative DR: 700

Tuy nhiên, phương pháp này lại bỏ đi một lượng lớn dữ liệu, từ 35 000 dữ liệu xuống còn 3500. Bài toán cần phát hiện điểm xuất huyết, điểm phình to của mạch máu,… rất khó phát hiện, vậy việc 31 500 dữ liệu bị bỏ đi là một sự lãng phí lớn trong quá trình đào tạo mô hình.

- **Sử dụng trọng số lớp:**

Phương pháp đánh trọng số lớp sử dụng được tối đa bộ dữ liệu so với phương pháp undersampling và dữ liệu không bị nhân bản lên nhiều lần, hạn chế được tình trạng overfitting so với phương pháp oversampling. Dưới đây là cách tạo ra bộ trọng số cho dữ liệu của bài toán:

```
No_DR = 17295
Mild =3581
Moderate = 1641
Severe = 605
Proliferative_DR = 470
total=No_DR + Mild + Proliferative_DR + Severe + Moderate
```

```
weight_for_Mild = (1 / Mild) * (total / 5.0)
weight_for_Moderete = (1 / Moderate) * (total / 5.0)
weight_for_NoDR = (1 / No_DR) * (total / 5.0)
weight_for_ProDR = (1 / Proliferative_DR) * (total / 5.0)
weight_for_Severe = (1 / Severe) * (total / 5.0)

class_weight = {0: weight_for_Mild, 1: weight_for_Moderate, 2:
weight_for_NoDR, 3: weight_for_ProDR, 4: weight_for_Severe}
```

Kết quả:

Nhãn	Trọng số
Mild	1.3176207763194636
Moderate	2.8753199268738574
No DR	0.2728187337380746
Proliferative DR	10.039148936170212
Severe	7.799008264462809

Bảng 2. 1 Trọng số lớp

2.3 Phân chia dữ liệu

2.3.1. Tập huấn luyện và tập kiểm thử

Dữ liệu được chia thành 2 tập, Bảng 2.2 dưới đây mô tả số lượng dữ liệu của từng nhãn trong tập huấn luyện và kiểm thử.

Nhãn	Tập huấn huyện	Tập kiểm thử
0	20525	5212
1	1955	481
2	4275	1023
3	760	159
4	570	147

Bảng 2. 2 Số lượng dữ liệu của từng nhãn trong tập huấn luyện và kiểm thử

Tập kiểm thử sẽ không được sử dụng trong quá trình tinh chỉnh tham số và đào tạo mà để kiểm tra đánh giá độ tin cậy ở bước cuối cùng.

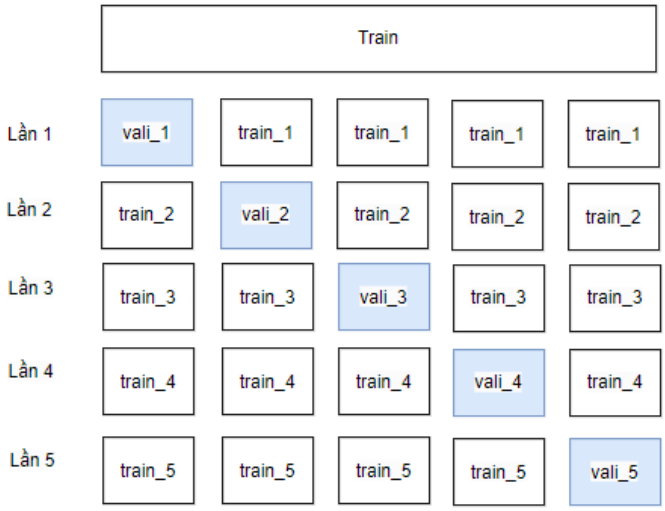
2.3.2. K-fold

Để đảm bảo dữ liệu huấn luyện không bị giảm và mang tính may rủi, khóa luận áp dụng phương pháp k-fold. Chia tập huấn luyện ta thu được ở mục 2.3.1 chia thành 5 fold. Bảng 2.3 dưới đây mô tả số lượng dữ liệu của một lần đào tạo.

Nhãn	Tập huấn huyện	Tập xác thực
0	16454	4105
1	1580	319
2	3424	855
3	561	152
4	449	114

Bảng 2. 3 Số lượng dữ liệu của từng nhãn trong một lần đào tạo

- Tỷ lệ dữ liệu giữa các nhãn trong các fold cần đảm bảo phải tương tự nhau. Dữ liệu trong tập xác thực không trùng với dữ liệu trong tập huấn luyện.



Hình 2. 6. Dữ liệu để đào tạo mô hình

- Khi triển khai đào tạo cho các mô hình CNN, tập huấn luyện được chia thành 5 phần để thực hiện CV. *Hình 2.6* mô tả tập dữ liệu huấn luyện của từng mô hình.

Trong quá trình đào tạo các mô hình lần lượt được huấn luyện năm lần với bốn phần của tập huấn luyện và một phần còn lại của tập xác thực. Để tận dụng các hình ảnh được thu thập và giảm khả năng khớp quá mức, khóa luận đã sử dụng các phương pháp tăng cường dữ liệu như cắt, lật và xoay ngẫu nhiên trên dữ liệu huấn luyện.

2.3.3. Chia dữ liệu vào thư mục gán nhãn

Để thuận thiện cho việc huấn luyện mô hình dữ liệu cần được chia vào các thư mục nhãn tương ứng. Nhãn của các ảnh đã được các bác sĩ phân loại trong file *trainLabels_cropped.csv.* File bao gồm cột tên ảnh và mức độ của bệnh được đánh số từ 0 đến 4. Hình 2.7 dưới đây là cấu trúc file *trainLabels_cropped.csv:*

	Unnamed: 0	image	level
0	0	10_left	0
1	1	10_right	0
2	2	13_left	0
3	3	13_right	0
4	4	15_left	1
5	5	15_right	2
6	6	16_left	4
7	7	16_right	4
8	8	17_left	0
9	9	17_right	1
10	10	19_left	0

Hình 2. 7 Cấu trúc bảng nhãn của dữ liệu

Thực hiện chia các ảnh vào 5 thư mục tương ứng với 5 nhãn dưới dây:

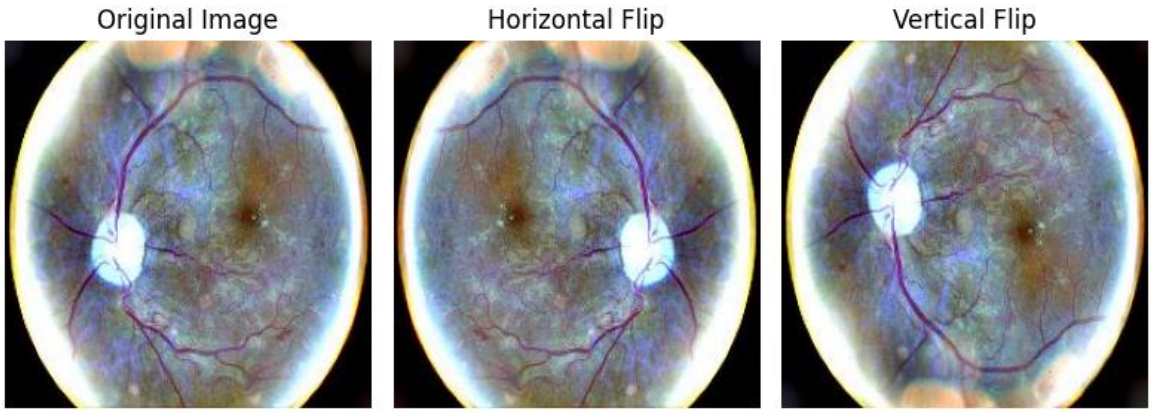
- Nếu level = 0 ảnh sẽ được di chuyển vào thư mục nhãn No DR
- Nếu level = 1 ảnh sẽ được di chuyển vào thư mục nhãn Mild
- Nếu level = 2 ảnh sẽ được di chuyển vào thư mục nhãn Moderate
- Nếu level = 3 ảnh sẽ được di chuyển vào thư mục nhãn Severe
- Nếu level = 4 ảnh sẽ được di chuyển vào thư mục nhãn Proliferative DR

2.4 Tăng cường dữ liệu

Để cải thiện khả năng tổng quát hóa và hiệu suất của mô hình, bài toán đã áp dụng các kỹ thuật tăng cường dữ liệu bao gồm lật ngang và lật dọc cho dữ liệu huấn luyện. Khi sử dụng ImageDataGenerator từ thư keras.preprocessing.image, mỗi hình ảnh trong tập huấn luyện sẽ được áp dụng ngẫu nhiên các biến đổi này như lật ngang và lật dọc.

```
train_image_generator = ImageDataGenerator(horizontal_flip=True,
vertical_flip=True, rescale=1./255)
```

Tuy nhiên, thay vì tạo ra và lưu trữ các phiên bản tăng cường của từng hình ảnh, ImageDataGenerator chỉ tạo ra các biến thể của từng hình ảnh trong thời gian thực khi chúng được truy xuất từ thư mục đầu vào. Do đó, tổng số lượng hình ảnh vẫn giữ nguyên so với tập dữ liệu gốc vì ImageDataGenerator không tạo ra các bản sao của hình ảnh mà chỉ áp dụng các biến đổi tăng cường vào từng hình ảnh khi chúng được lấy ra từ thư mục đầu vào để huấn luyện, ImageDataGenerator sẽ thực hiện các thay đổi theo các tham số đã cấu hình trước. Quá trình này giúp tăng tính đa dạng của dữ liệu đầu vào, từ đó giảm thiểu hiện tượng overfitting, cải thiện khả năng tổng quát hóa của mô hình. Điều này là một phần quan trọng trong quá trình phát triển mô hình học sâu để đạt được kết quả phân loại chính xác và ổn định.



Hình 2. 8 Ảnh võng mạc trước và sau khi tăng cường dữ liệu

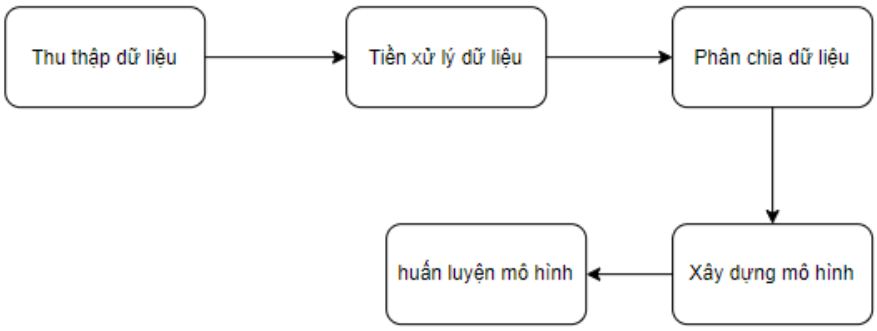
CHƯƠNG 3: XÂY DỰNG VÀ TINH CHỈNH MÔ HÌNH

Trong phần này, khóa luận sẽ nêu các mô hình mạng CNN được sử dụng cho bài toán, đánh giá kết quả và tinh chỉnh mô hình.

3.1 Mô tả bài toán

Mục tiêu của khóa luận này là thực hiện thu thập dữ liệu, xử lý dữ liệu, sau đó sẽ xây dựng, huấn luyện và đánh giá mô hình. Để làm điều này, khóa luận cũng thực hiện một số phương pháp sử dụng tiền huấn luyện mô hình VGG-16, VGG19, Resnet-50, Resnet-101 để tiết kiệm thời gian huấn luyện mô hình. Mô hình cần phát hiện ra những đặc điểm của từng giai đoạn bệnh như phình mạch máu, điểm xuất huyết, … để đưa ra dự đoán giai đoạn mắc bệnh. Cuối cùng, khoá luận cũng sử dụng một số phương pháp tăng cường dữ liệu và ghép nối các mô hình nhằm cải thiện độ chính xác.

Hình 3.1 dưới đây sẽ mô toàn toàn bộ quá trình thực hiện khóa luận này:



Hình 3. 1. Quy trình thực hiện khóa luận

Các phần tiếp theo sẽ mô tả chi tiết hơn về các bước

3.2 Xây dựng mô hình

Trước khi tiến hành đào tạo mô hình và tinh chỉnh tham số cần phải lựa chọn ra kiến trúc phù hợp. Tập dữ liệu sử dụng cho việc xây dựng mô hình là 1 phần nhỏ lấy ra từ tập dữ liệu gốc với 300 mẫu cho mỗi nhãn. Tổng số lượng mẫu cho tập huấn luyện là 1500 mẫu

3.2.1. Kiến trúc sử dụng

- Lớp đầu vào có kích thước (224,224,3)
- Lớp đầu ra có 5 nút với hàm kích hoạt là softmax
- Hàm tối ưu sử dụng là Adam
- Hàm mất mát sử dụng là Categorical Crossentropy

Model	Lớp ẩn	Epochs	Batchsize
VGG-16	- Lớp drop out với rate 0.5	300	32
VGG-19	- Lớp drop out với rate 0.5	300	32
VGG-19_2	- Lớp dense gồm 256 nút - Lớp dense gồm 128 nút - Lớp dense gồm 64 nút - Lớp dense gồm 32 nút - Lớp drop out với rate 0.5	300	32
Resnet-50	- Lớp drop out với rate 0.5	300	32
Resnet-101	- Lớp drop out với rate 0.7	300	32

Bảng 3. 1: Kiến trúc của các mô hình CNN được sử dụng

3.2.2. Kết quả

Bảng 3.2 dưới đây mô tả về độ chính xác trên tập huấn luyện để lựa chọn ra kiến trúc phù hợp.

Model	Kết quả sai số	Kết quả độ chính xác
VGG-16	1.608	21,34%
VGG-19	1.608	21,34%
VGG19_2	0.054	98,50%
Resnet-50	0.125	95,66%
Resnet-101	0.231	92,01%

Bảng 3. 2: Kết quả của các mô hình CNN

Nhận xét:

- Sử dụng kiến trúc mô hình VGG-16, VGG-19 cho kết quả độ chính xác thấp.
- Mô hình Resnet-50, Resnet-101 và VGG-19_2 (sau khi thêm 4 layer dense) cho kết quả với tập huấn luyện tốt, đều trên 90%

Kết luận: loại bỏ mô hình VGG-16 và VGG-19, tiếp tục cải tiến và đánh giá trên tập xác thực với 3 mô hình Resnet-50, Resnet-101 và VGG-19_2

3.3 Tinh chỉnh tham số

3.3.1. VGG-19

- Lớp đầu vào có kích thước (224,224,3)
- Lớp ẩn:
    - Các lớp cũ của VGG-19
    - Lớp dense gồm 256 nút
    - Lớp dense gồm 128 nút
    - Lớp dense gồm 64 nút

    - Lớp dense gồm 32 nút
    - Lớp drop out với rate 0.5
- Lớp đầu ra có 5 nút với hàm kích hoạt là softmax
- Hàm tối ưu sử dụng là Adam
- Hàm mất mát sử dụng là Categorical Crossentropy
- Đóng băng 50% layer đầu của mô hình

Model	Learning rate	Epochs	Batch size	Dữ liệu
Model_1	0.001	100	32	Chưa tăng cường
Model_2	0.0001	100	32	Chưa tăng cường
Model_3	0.00001	100	32	Chưa tăng cường
Model_4	0.001	100	64	Đã tăng cường
Model_5	0.0001	100	64	Đã tăng cường
Model_6	0.00001	100	64	Đã tăng cường

Bảng 3. 3: Siêu tham số được sử dụng trong VGG-19

Bảng 3.4 dưới đây mô tả kết quả về độ chính xác trên tập huấn luyện, tập xác thực trên từng fold.

Model	Kết quả độ chính xác									
	Tập huấn luyện					Tập xác thực				
	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Model_1	88.43%	88.27%	80.98%	83.19%	86.41%	68.59%	65.84%	65.69%	66.73%	68.89%
Model_2	89.61%	88.55%	85.21%	86.89%	86.42%	69.23%	68.77%	67.90%	67.27%	69.55%
Model_3	84.43%	82.21%	80.75%	80.05%	84.63%	67.29%	67.37%	62.39%	62.00%	64.76%
Model_4	94.17%	95.24%	95.76%	93.20%	94.20%	78.23%	77.87%	76.48%	78.12%	78.17%
Model_5	99.16%	97.48%	98.72%	99.33%	98.36%	81.29%	79.64%	80.82%	79.86%	80.49%
Model_6	95.18%	96.84%	96.57%	95.35%	96.66%	78.23%	78.87%	78.50%	77.92%	78.12%

Bảng 3. 4: Kết quả độ chính xác của mô hình VGG-19

Nhận xét:

- Hiệu suất đào tạo và kiểm tra: Các mô hình đều có kết quả đào tạo (tập huấn luyện) cao hơn so với kiểm tra (tập xác thực), cho thấy sự khác biệt giữa đào tạo và tổng quát hóa.
- Mô hình tốt nhất: Model_5 có xu hướng cho kết quả tốt nhất với độ chính xác cao nhất trên tập xác thực và tập huấn luyện.

Kết luận:

- Để đạt được kết quả tốt nhất, cần phải điều chỉnh và tối ưu hóa các siêu tham số này một cách cẩn thận dựa trên hiệu suất trên tập xác thực.
- Chọn siêu tham số cho mô hình VGG-19:
    - Learning rate = 0.0001.
    - Batch size = 64.

3.3.2. Resnet-50

- Lớp đầu vào có kích thước (224,224,3)
- Lớp đầu ra có 5 nút với hàm kích hoạt là softmax
- Hàm tối ưu sử dụng là Adam
- Hàm sai số sử dụng là Categorical Crossentropy
- Đóng băng 70% mô hình

Model	Lớp ẩn	Learning rate	Epochs	Dữ liệu	Dữ liệu
Model_1	- Lớp drop out với rate 0.5	0.0001	100	32	Chưa tăng cường
Model_2	- Lớp drop out với rate 0.5 - Lớp dense gồm 256 nút - Regularizers l2 ở các lớp dense	0.0001	100	32	Chưa tăng cường
Model_3	- Lớp drop out với rate 0.5	0.00001	100	32	Đã tăng cường

Model_4	- Lớp drop out với rate 0.5	0.001	100	64	Đã tăng cường
Model_5	- Lớp drop out với rate 0.5	0.0001	100	64	Đã tăng cường

Bảng 3. 5: Siêu tham số được sử dụng trong Resnet-50

Bảng 3.6 dưới đây mô tả kết quả về độ chính xác trên tập huấn luyện, tập xác thực trên từng fold.

Model	Kết quả độ chính xác									
	Tập huấn luyện					Tập xác thực				
	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Model_1	90.22%	89.93%	90.66%	91.72%	88.81%	47.52%	50.38%	48.52%	46.11%	50.36%
Model_2	98.64%	97.05%	98.55%	98.58	97.30%	60.45%	61.04%	63.22%	61.87%	64.01%
Model_3	98.10%	97.28%	96.78%	96.27%	97.47%	76.76%	74.83%	74.15%	75.54%	76.34%
Model_4	97.20%	96.27%	96.78%	97.02%	95.79%	78.22%	75.54%	74.15%	76.24%	74.56%
Model_5	98.95%	97.21%	98.68%	97.20%	98.93%	81.41%	79.09%	80.82%	83.87%	79.51%

Bảng 3. 6: Kết quả độ chính xác của mô hình Resnet-50

Nhận xét:

- Sử dụng Regularizers l2 nhưng không tăng cường dữ liệu vẫn không giảm khả năng model bị overfit trong bài toán này.
- Mô hình tốt nhất: Model_5 có kết quả cao trên cả tập huấn luyện và tập xác thực. Điều này cho thấy sự hiệu quả của các siêu tham số đã chọn và sự quan trọng của việc tăng cường dữ liệu

Kết luận:

- Các siêu tham số như tỷ lệ học tập và kích thước lô huấn luyện có ảnh hưởng đáng kể đến hiệu suất của mô hình ResNet-50.
- Chọn siêu tham số cho mô hình Resnet-50:
    - Learning rate = 0.0001.
    - Batch size = 64.

3.3.3. Resnet-101

- Lớp đầu vào có kích thước (224,224,3)
- Lớp đầu ra có 5 nút với hàm kích hoạt là softmax
- Hàm tối ưu sử dụng là Adam
- Hàm sai số sử dụng là Categorical Crossentropy
- Đóng băng 70% mô hình

Model	Lớp ẩn	Learning rate	Epochs	Batch size	Dữ liệu
Model_1	- Lớp drop out với rate 0.5	0.00001	100	32	Chưa tăng cường
Model_2	- Lớp drop out với rate 0.5 - Lớp dense gồm 256 nút	0.00001	100	32	Chưa tăng cường

	- Regularizers l2 ở các lớp dense				
Model_3	- Lớp drop out với rate 0.5	0.0001	100	64	Đã tăng cường

Bảng 3. 7: Siêu tham số được sử dụng trong Resnet-101

Bảng 3.8 dưới đây mô tả kết quả về độ chính xác trên tập huấn luyện, tập xác thực trên từng fold.

Model	Kết quả độ chính xác									
	Tập huấn luyện					Tập xác thực				
	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Model_1	93.52%	92.73%	92.98%	92.73%	93.97%	45.38%	47.74%	49.58%	45.97%	46.35%
Model_2	98.95%	96.62%	95.26%	97.97%	96.37%	65.76%	62.17%	63.98%	64.73%	63.86%
Model_3	96.43%	95.75%	98.31%	99.58%	99.50%	73.90%	76.17%	75.95%	76.75%	76.29%

Bảng 3. 8: Kết quả độ chính xác của mô hình Resnet-101

Nhận xét:

- Model_1 có độ chính xác trên tập huấn luyện cao hơn rất nhiều so với tập xác thực, cho thấy mô hình có xu hướng overfitting.
- Model_2 có sử dụng Regularizers l2 nhưng kết quả tập xác thực vẫn không tăng nhiều.

- Model_3 có xu hướng cải thiện so với hai model trước đó trên cả tập huấn luyện và tập xác thực cho thấy vai trò của batch size và tăng cường dữ liệu.

Kết luận:

- Các siêu tham số như tỷ lệ học tập và kích thước lô huấn luyện có ảnh hưởng đáng kể đến hiệu suất của mô hình ResNet-101.
- Chọn siêu tham số cho mô hình Resnet-101:
  - Learning rate = 0.0001.
  - Batch size = 64.

3.4 Huấn luyện các mô hình

Sau khi chọn được các siêu tham số cho các mô hình thì cần huấn luyện các mô hình bằng tập huấn luyện và sử dụng tập kiểm thử để đánh giá. Bảng 3.9 dưới đây mô tả kết quả về độ chính xác trên tập huấn luyện, tập xác thực của các mô hình,

Model	Tập huấn luyện		Tập kiểm thử	
	Sai số	Độ chính xác	Sai số	Độ chính xác
VGG-19_2	0.1100	95.94	0.8354	81.15
Resnet-50	0.1158	95.84	0.5909	81.08
Resnet-101	0.0284	99.20	1.1817	74.85

Bảng 3. 9: Kết quả sai số và độ chính xác của các mô hình

Nhận xét:

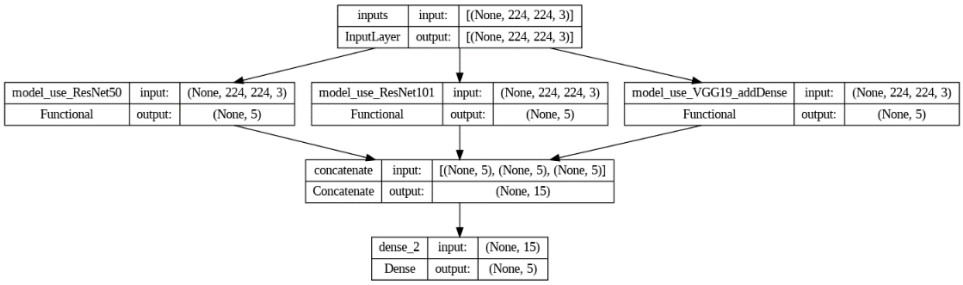
- Các mô hình VGG-19_2, Resnet-50 và Resnet-101 đều cho thấy kết quả khả quan trên cả tập huấn luyện và tập kiểm thử.
- Các mô hình VGG-19_2, Resnet-50 có khả năng khái quát hoá tốt. Mô hình Resnet-101 học tập rất tốt nhưng mô hình có dấu hiệu overfitting nhẹ khi độ chính xác giảm trên tập kiểm thử.

Kết luận: Sử dụng cả ba mô hình vào phương pháp “ghép nối các mô hình”.

3.5 Ghép nối các mô hình

Phương pháp ghép nối mô hình là một phương pháp kết hợp các mô hình khác nhau để tạo thành một mô hình lớn hơn, mạnh mẽ hơn hoặc giải quyết nhiều nhiệm vụ. Ý tưởng của việc combine các mô hình khác nhau xuất phát từ một suy nghĩ hợp lý là: các mô hình khác nhau có khả năng khác nhau, có thể thực hiện tốt nhất các loại công việc khác nhau, khi kết hợp các mô hình này với nhau một cách hợp lý thì sẽ tạo thành một mô hình kết hợp (combined model) mạnh có khả năng cải thiện hiệu suất tổng thể (overall performance) so với việc chỉ dùng các mô hình một cách đơn lẻ.

Trong bài toán này, khóa luận kết hợp ba mô hình khác nhau: VGG-19_2, Resnet-50 và Resnet-101. VGG-19_2 có khả năng học và tổng quát hoá tốt trên dữ liệu huấn luyện, trong khi Resnet-50 thể hiện sự ổn định và độ chính xác trên cả tập huấn luyện và tập kiểm thử. Trong khi đó, Resnet-101 cho thấy khả năng học tập mạnh mẽ nhưng có dấu hiệu overfitting trên tập kiểm thử. Bằng cách kết hợp các dự đoán từ ba mô hình này, ensemble learning có thể giúp giảm thiểu những nhược điểm riêng biệt của từng mô hình và mang lại dự đoán chính xác hơn cho bài toán xác định. Hình 3.2 dưới đây sẽ mô tả chi tiết cấu trúc của phương pháp này



Hình 3. 2 Sơ đồ ghép nối các mô hình

Kết quả: Sau khi áp dụng ensemble learning từ ba mô hình, kết quả huấn luyện và kiểm thử cho thấy hiệu suất dự đoán đã được cải thiện đáng kể. Trên tập huấn luyện, độ chính xác đạt 99.78%, chỉ số này cho thấy mô hình tổ hợp đã học và khái quát hóa tốt trên dữ liệu huấn luyện. Đối với tập kiểm thử, độ chính xác là 85.21%, cho thấy mô hình tổ hợp đã giảm thiểu được overfitting và đem lại dự đoán chính xác và ổn định trên dữ liệu mới. Kết quả này cho thấy sự hiệu quả của việc sử dụng ensemble learning trong việc cải thiện hiệu suất của hệ thống dự đoán so với việc sử dụng một mô hình đơn lẻ.

CHƯƠNG 4: MÔ PHỎNG VÀ ĐÁNH GIÁ

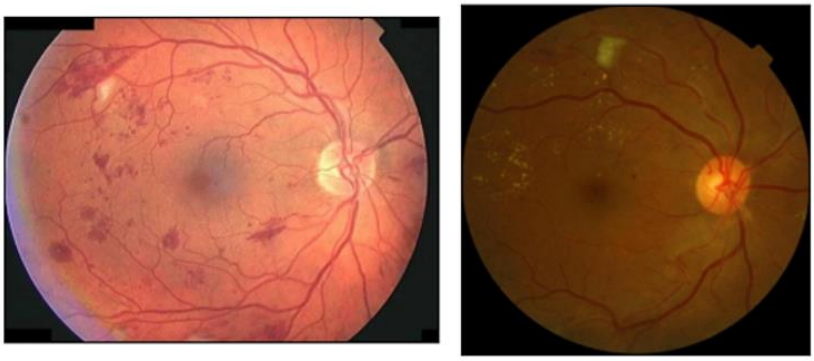
Chương này sẽ mô phỏng về kết quả dự đoán và đánh giá cuối cùng về phương pháp dự đoán nghiên cứu trong bài.

4.1 Mô phỏng

Sử dụng lại trọng số mà chúng ta đã huấn luyện để thực hiện dự đoán những bức ảnh. Để có thể dự đoán xem ảnh chụp võng mạc của bệnh nhân có mắc võng mạc đái tháo đường hay không và đang ở giai đoạn nào thì cần đưa ảnh về kích thước 224x224 pixel, sau đó thực hiện các bước tiền xử lý để làm nổi bật những chi tiết như mạch máu, điểm xuất huyết,… Thực hiện đưa bức ảnh đã được tiền xử lý bên trên vào mô hình để thực hiện dự đoán.

Thử nghiệm 1:

Dự đoán hai ảnh chụp võng mạc sau:



Hình 4. 1 Ảnh chụp võng mạc cho thử nghiệm 1

Hình 4.2 dưới đây là kết quả dự đoán của thử nghiệm 1 khi sử dụng kết quả đào tạo mô hình mà khóa luận làm được:

Hình 4. 2 Kết quả dự đoán cho thử nghiệm 1

Mô hình đã dự đoán chính xác giai đoạn của bệnh với 2 trường hợp:

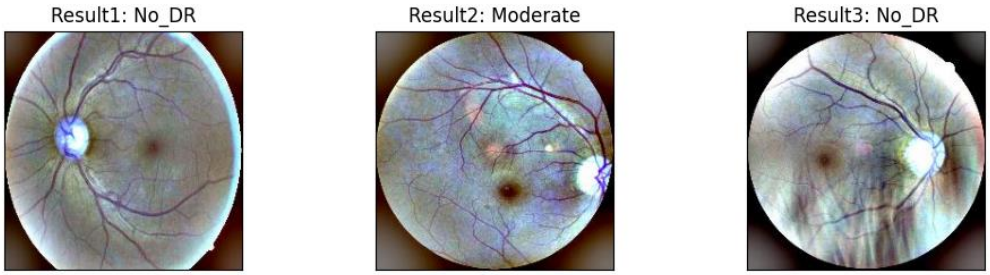
- Result 1 là ảnh chụp võng mạc rõ, nét, ánh sáng tốt, những dấu hiện nhận biết cũng rất rõ ràng. Mô hình đã dự đoán đúng bệnh nhân đang ở giai đoạn nặng nhất - giai đoạn tăng sinh (Proliferative DR).
- Result 2 là ảnh chụp võng mạc có điều kiện ánh sáng kém, những dấu hiệu nhận biết cũng rất khó thấy hơn so với Result 1. Tuy nhiên, mô hình vẫn dự đoán tốt bệnh nhân đang ở giai đoạn tiền tăng sinh (Severe).

Thử nghiệm 2:

Dự đoán 3 ảnh chụp võng mạc sau:

Hình 4. 3 Ảnh chụp võng mạc cho thử nghiệm 2

Hình 4.4 dưới đây là kết quả dự đoán của thử nghiệm 2 khi sử dụng kết quả đào tạo mô hình mà khóa luận làm được:



Hình 4. 4 Kết quả dự đoán cho thử nghiệm 2

Mô hình đã dự đoán 2 trường hợp đúng là result 1, result 2 và 1 trường hợp sai là result 3:

- Result 1: Ảnh chụp võng mạc thiếu ánh sáng, tuy nhiên, khi tăng cường ánh sáng thì mạch máu hiện rõ vậy nên mô hình có thể dự đoán kết quả bệnh nhân không mắc bệnh.
- Result 2: Ảnh chụp võng mạc thiếu ánh sáng, tuy nhiên, khi tăng cường ánh sáng thì mạch máu, những điểm xuất huyết hiện rõ hơn vậy nên mô hình có thể dự đoán được bệnh nhân đang ở giai đoạn 2 - bệnh lý hoàng điểm do đái tháo đường (Moderate).

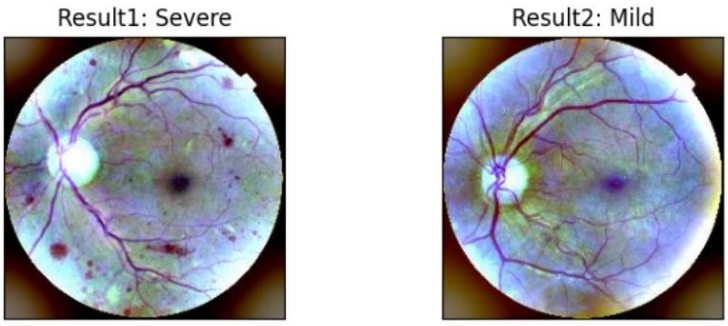
- Result 3: Ảnh chụp võng mạc thiếu ánh sáng và có những vệt đen. Khi tăng cường ánh sáng thì mạch máu hiện rõ hơn nhưng những vệt đã che một phần bức ảnh, ở giai đoạn 1 thì những dấu hiệu của bệnh chưa rõ ràng vậy nên mô hình đã dự đoán sai bệnh nhân này không mắc bệnh nhưng ảnh chụp võng mạc của bệnh nhân này đang ở giai đoạn 1 - Bệnh lý võng mạc nền (Mild).

Thử nghiệm 3:

Dự đoán hai ảnh chụp võng mạc sau:

Hình 4. 5 Ảnh chụp võng mạc cho thử nghiệm 3

Hình 4.6 dưới đây là kết quả dự đoán của thử nghiệm 3 khi sử dụng kết quả đào tạo mô hình mà khóa luận làm được:



Hình 4. 6 Kết quả dự đoán cho thử nghiệm 3

Mô hình đã dự đoán chính xác giai đoạn của bệnh với 2 trường hợp:

- Result 1: Ảnh chụp võng mạc có ánh sáng tốt, khi tăng cường ánh sáng thì mạch máu, các điểm xuất huyết hiện rõ vậy nên mô hình có thể dự đoán kết quả bệnh nhân đang ở giai đoạn tiền tăng sinh (Severe).
- Result 2: Ảnh chụp võng mạc có ánh sáng tốt, khi tăng cường ánh sáng thì mạch máu hiện rõ điểm phình vậy nên mô hình có thể dự đoán được bệnh nhân đang ở giai đoạn 1 - Bệnh lý võng mạc nền (Mild)

4.2 Đánh giá

Trong phần này, khóa luận sẽ đánh giá hiệu quả và độ chính xác của phương pháp ứng dụng trí tuệ nhân tạo để phát hiện bệnh võng mạc đái tháo đường.

Đầu tiên, khóa luận sẽ đánh giá kết quả của việc thu thập và xử lý dữ liệu. Quá trình thu thập dữ liệu đã được thực hiện như kế hoạch và đáp ứng được yêu cầu của nghiên cứu. Dữ liệu đã được tiền xử lý và phân chia thành tập huấn luyện và tập kiểm tra một cách chính xác.

Tiếp theo, khóa luận sẽ đánh giá hiệu suất của mô hình học máy được xây dựng. Khóa luận sử dụng một số mạng CNN như VGG, Resnet và điều chỉnh các siêu tham số để đạt được kết quả tốt nhất. Kết quả của mô hình được đánh giá bằng độ chính xác.

Kết quả đánh giá cho thấy phương pháp ghép nối mô hình đạt được độ chính xác khá cao 85% và khả năng phát hiện bệnh võng mạc đái tháo đường khá tốt. So với các phương pháp khác, phương pháp ghép nối mô hình có độ tin cậy cao hơn việc sử dụng một mô hình dự đoán.

	precision	recall	f1-score	support
Mild	1.00	0.73	0.84	11
Moderate	0.94	0.94	0.94	18
No_DR	0.85	1.00	0.92	29
Proliferative_DR	1.00	1.00	1.00	24
Severe	1.00	0.89	0.94	18
accuracy			0.94	100
macro avg	0.96	0.91	0.93	100
weighted avg	0.95	0.94	0.94	100

Hình 4. 7 Đánh giá độ chính xác trên từng nhãn

Mô hình học máy đã đạt được độ chính xác cao và khả năng phát hiện tốt trên từng lớp. Tuy nhiên, cần tiếp tục nghiên cứu và cải tiến để đảm bảo tính ổn định và độ tin cậy của phương pháp trong các tình huống thực tế.

KẾT LUẬN

Ở chương cuối cùng, tài liệu này sẽ đưa ra các kết quả và những kiến thức thu đã được. Đồng thời, khóa luận cũng đề xuất một số định hướng cho tương lai để có thể tiếp tục cải tiến và hoàn thiện đề tài.

1. Tổng kết

Qua quá trình thực hiện khóa luận này, chúng tôi đã biết thêm được các kiến thức về xử lý dữ liệu ảnh. Ngoài ra, chúng tôi còn nắm vững được các kiến thức về xây dựng mô hình học máy, CNN, tìm hiểu được thêm nhiều mô hình sử dụng cho phân loại hình ảnh và biết được thêm nhiều phương pháp xử lý, tăng cường dữ liệu.

Mục tiêu của nghiên cứu này trước hết là thu thập và xử lý dữ liệu thu thập được, tiếp đến tìm hiểu và xây dựng các mô hình mạng CNN giúp giải quyết các vấn đề của bài toán phát hiện bệnh võng mạc đái tháo đường, phân lạo xem bệnh đang ở giai đoạn nào để hỗ trọ các bác sĩ đưa ra phương pháp điều trị kịp thời. Các mô hình được chúng tôi sử dụng cho khóa luận này gồm 4 mô hình VGG-16, VGG-19, Resnet-50, Resnet101. Ngoài ra, qua nghiên cứu này chúng tôi còn biết sử dụng tiền huấn luyện của các mô hình để hỗ trợ cho quá trình đào tạo mô hình.

Thời gian thực hiện khóa luận này không nhiều nên dữ liệu và thời gian đào tạo mô hình dự đoán còn giới hạn. Tuy nhiên bằng cách tìm tòi các phương pháp, khóa luận đã rút ra một số kết quả hữu ích có thể là định hướng kinh nghiệm cho các nghiên cứu trên chủ đề này về sau. Cụ thể là, thông qua kết quả, khóa luận rút ra được cần có thêm dữ liệu, phương pháp xử lý ảnh. Thử nghiệm cho thấy mô hình dự đoán đạt kết quả kém với những mô hình mạng CNN quá đơn giản như VGG16 hay VGG19 đã được tiền huấn luyện. Độ chính xác đạt kết quả cao hơn khi sử dụng những mô hình phức tạp như Resnet50, Resnet101, và VGG19 khi thêm các layer. Cuối cùng, việc kết hợp 3 mô hình cho kết quả khả quan hơn so với việc chỉ sử dụng 1 mô hình để dự đoán.

2. Định hướng

Từ kết quả đạt được, trong tương lai, chúng tôi muốn thử nghiệm thực hiện thêm một số phương pháp tăng cường dữ liệu khác và tìm thêm các nguồn dữ liệu bổ sung cho tập dữ liệu hiện có. Cuối cùng, chúng tôi dự định tìm kiếm các phương pháp huấn luyện và học tăng cường tốt hơn để mô hình học máy có khả năng tự cập nhật từ đó có thể cải tiến chất lượng.

TÀI LIỆU THAM KHẢO SÁCH VÀ BÁO

[1] "Cổng thông tin Bộ Y tế," 13 11 2022. [Online]. Available: https://moh.gov.vn/tin-noi-bat/-/asset_publisher/3Yst7YhbkA5j/content/khoang-5-trieu-nguoi-viet-ang-mac-can-benh-gay-nhieu-bien-chung-ve-tim-mach-than-kinh-cat-cut-chi-.

[2] "Vinmec," 01 06 2024. [Online]. Available: https://www.vinmec.com/vi/tin-tuc/thong-tin-suc-khoe/benh-vong-mac-dai-thao-duong/.

[3] T. M. Mitchell and T. M. Mitchell, Machine learning, vol. 1, McGraw-hill New York,, 1997.

[4] Y. B. a. G. H. Y.Lecun, Deep Learning, 2015.

[5] "Giới thiệu về CNN," 14 03 2024. [Online]. Available: https://www.geeksforgeeks.org/introduction-convolution-neural-network/.

[6] "Kiến trúc VGG-16," 21 03 2024. [Online]. Available: https://www.geeksforgeeks.org/vgg-16-cnn-model/.

[7] "Kiến trúc VGG-19," 24 03 2023. [Online]. Available: https://vinbigdata.com/kham-pha/04-mo-hinh-pre-trained-cnn-giup-ban-giai-quyet-cac-bai-toan-thi-giac-may-tinh-voi-transfer-learning.html.

[8] "mô tả resnet-50," 01 06 2024. [Online]. Available: https://www.mathworks.com/help/deeplearning/ref/resnet50.html.

[9] "Kiến trúc Resnet-50," 01 06 2024. [Online]. Available: https://miro.medium.com/v2/resize:fit:1400/0*9LqUp7XyEx1QNc6A.png.

[10] "mô tả Resnet-101," 01 06 2024. [Online]. Available: https://www.mathworks.com/help/deeplearning/ref/resnet101.html.

[11] "Kiến trúc Resnet-101," 01 06 2024. [Online]. Available: https://www.mdpi.com/mathematics/mathematics-11-00841/article_deploy/html/images/mathematics-11-00841-g002.png.

[12 "drop-out," 2024. [Online]. Available: https://images.viblo.asia/full/363449ef-
] bc8e-4476-98b9-a450498aa5f6.jpeg.

[13 "Giới thiệu hàm kích hoạt," 01 06 2024. [Online]. Available:
] https://www.geeksforgeeks.org/activation-functions-neural-networks/.

[14 "Hàm kích hoạt sigmoid," 06 2024. [Online]. Available:
] https://media.geeksforgeeks.org/wp-content/uploads/20221013120722/1.png.

[15 "Hàm kích hoạt softmax," 06 2024. [Online]. Available:
] https://media.geeksforgeeks.org/wp-content/uploads/20190322133046/softmxx.png.

[16 D. P. K. a. J. Ba, “Adam: A method for stochastic optimization,”, ArXiv Prepr.
] ArXiv14126980, 2014.

[17 "Cách hoạt động của k-fold CV," 06 2024. [Online]. Available: https://scikit-
] learn.org/stable/modules/cross_validation.html.

[18 "Phương pháp OverSampler," 2024. [Online]. Available:
] https://media.licdn.com/dms/image/D5612AQGXHS4zGawTyQ/article-cover_image-shrink_720_1280/0/1670033275223?e=2147483647&v=beta&t=FWxVNYk2OO0cGauh7K2tM9-RjijnZQL3is6HqFa1gzg.

[19 "Phương pháp UnderSampler," 2024. [Online]. Available:
] https://miro.medium.com/v2/resize:fit:662/1*RfRHzqUI0753EjN35oI8Zg.png.

[20 "bệnh võng mạc đái tháo đường," 06 2024. [Online]. Available:
] https://www.vinmec.com/vi/tin-tuc/thong-tin-suc-khoe/benh-vong-mac-dai-thao-duong/.

[21 "Dấu hiệu của bệnh võng mạc đái tháo đường," 2024. [Online]. Available:
] https://www.vinmec.com/s3-images/size/xxlarge/20190523_082220_729069_benh-vong-mac-dai-t.max-1800x1800.jpg.