



ỨNG DỤNG CNN NHẬN BIẾT 7 LOẠI BỆNH UNG THƯ DA QUA HÌNH ẢNH VÀ VIDEO

Nguyễn Trọng Đại

19146146

TRÍ TUỆ NHÂN TẠO - LỚP 05CLC

Tóm tắt

Cơ sở và mục tiêu của đề tài: Ung thư da hay da bị tổn thương có dấu hiệu là những đốm đỏ trên tay, mụn nhỏ tưởng chừng như nốt ruồi hay những dấu hiệu được coi là bất thường có trên da người so với các khu vực xung quanh của da. Một vài dấu hiệu có thể vô hại như những vết xước nhỏ hoặc có thể nghiêm trọng hơn là ung thư da. Và để nhận biết được những vùng da bị tổn thương này, khám ở các bệnh viện thường khá là đắt đỏ vì thế trong bài báo này tôi sẽ trình bày về việc ứng dụng mạng CNN để sần lọc 7 loại bệnh về ung thư da.

Phương pháp: Trong bài báo này tôi xây dựng mạng CNN có khả năng phân loại được 7 loại ung thư da với độ chính xác chấp nhận được, và tập dữ liệu mà tôi sử dụng là HAM10000. Và tôi sử dụng Framework Flask kết hợp cùng TensorflowJS để xây dựng ứng dụng nhận diện thông qua Website.

Kết quả: Tôi đã xây dựng được mạng CNN để sần lọc với kết quả accuracy = 77%, recall = 46%, precision = 59%. Cùng với đó tôi đã xây dựng hệ thống Webserver nhận diện realtime trực tuyến tại địa chỉ: <https://skincancer.svute.com>.

Kết luận: Hi vọng với sản phẩm này có thể giúp người có nghi ngờ về bệnh của mình có thể sử dụng để sần lọc trước khi đến bệnh viện khám. Và giúp bác sĩ có thể sần lọc bệnh nhân và giảm chi phí khám bệnh.

1 Giới thiệu

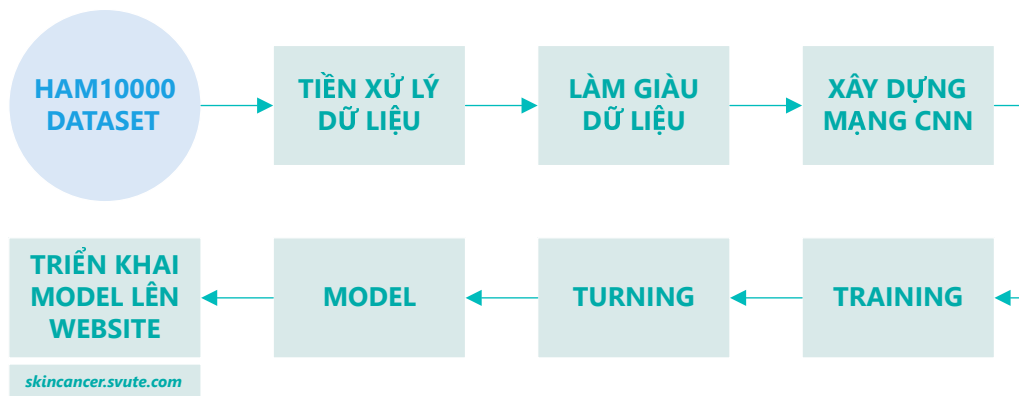
Ung thư da là một trong những bệnh rất phổ biến trên thế giới, cụ thể ở Mỹ mỗi năm có hơn 5 triệu ca mắc phải, mỗi năm có hơn 9000 người chết vì ung thư da. Điều trị ung thư da tốn chi phí rất lớn. Vì thế ung thư da trở thành một mối đe dọa lớn đến với cộng đồng. Tỷ lệ mắc ung thư da đã tăng lên hàng năm là một điều rất đáng báo động.

Trước đây ung thư da được chẩn đoán lâm sàng và không có bất kỳ sự hỗ trợ nào, và được đánh giá bằng mức độ kinh nghiệm của người bác sĩ. Điều này dẫn đến thiếu sự chính xác trong chẩn đoán. Trong những năm gần đây thì kỹ thuật nội soi đã bắt đầu phát triển và được ứng dụng vào trong việc chẩn đoán bệnh của bác sĩ.

Với những lý do trên cùng với sự phát triển của kỹ thuật nội soi đã cho chúng ta một tập dữ liệu về tổn thương da, cùng với đó là sự phát triển của AI và Deep learning chúng ta có thể tạo ra một mô hình chẩn đoán lâm sàng. Vì thế trong bài báo này tôi sẽ ứng dụng "mạng thần kinh tích chập" - CNN nhằm mục đích tạo ra một mô hình chẩn đoán và tạo ra một môi trường Realtime cho mọi người có thể vào và chẩn đoán lâm sàng.

2 Phương pháp

Trong đề tài này tôi sẽ trải qua 7 bước theo quy trình như hình dưới đây. Đầu tiên là tôi sử dụng tập dữ liệu HAM10000 sẽ được trình bày ở phần sau, tôi sẽ trải qua các bước xử lý dữ liệu, xử lý mất cân bằng trong tập dữ liệu, khi dữ liệu đã ổn định tôi xây dựng cấu trúc mạng CNN thực hiện việc training và turning để cho ra model tốt nhất có thể và triển khai realtime lên Website với Framework Flask.



Hình 1: Các bước thực hiện đề tài

2.1 Tập dữ liệu HAM10000

HAM10000 là tập dữ liệu được sử dụng trong đề tài này là bộ dữ liệu được công khai bởi đại học Harvard. Bộ dữ liệu bao gồm 10015 ảnh soi da để phục vụ tạo ra một tiêu chuẩn trong việc chẩn đoán các bệnh về tổn thương da. Tập dữ liệu HAM10000 được sử dụng trong cuộc thi ISIC 2018, trong tập dữ liệu có 7 lớp và 7 lớp này là 7 loại bệnh về tổn thương da cụ thể ở bảng dưới đây.

Loại bệnh	Nhãn
Actinic keratoses	0
Basal cell carcinoma	1
Benign keratosis-like lesions	2
Dermatofibroma	3
Melanocytic nevi	4
Malignant Melanoma	5
Vascular lesions	6

Bảng 1: Phân lớp các loại bệnh trong tập dữ liệu HAM10000



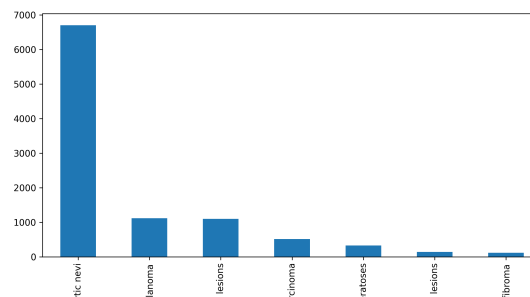
(a) Melanocytic nevi.

(b) Benign keratosis-like.

(c) Actinic keratoses.

Hình 2: Một vài hình ảnh trong tập HAM10000.

Tập dữ liệu HAM10000 bị vấn đề về mất cân bằng dữ liệu như hình dưới đây, trong tập dữ liệu này số lượng ảnh về bệnh "Melanocytic nevi" chiếm số lượng rất lớn



Hình 3: Phân bố dữ liệu tập HAM10000.

Phân lớp	0	1	2	3	4	5	6	Tổng
Số lượng hình ảnh	327	514	1099	115	6705	142	1113	10015

Bảng 2: Số lượng ảnh trong tập dữ liệu HAM10000

2.2 Tiền xử lý và làm giàu dữ liệu

Như đã trình bày ở phần trước thì tập HAM10000 bị tình trạng mất cân bằng rất lớn vì thế ở bước tiền xử lý dữ liệu tôi đã sử dụng phương pháp Data Augment nhằm mục đích làm giàu dữ liệu để cân bằng lại tập dữ liệu.

Trước khi xử lý dữ liệu mất cân bằng bằng Data Augment thì tôi sẽ áp dụng thêm một phương pháp để tăng độ nhạy của mô hình đối với phân lớp gây mất cân bằng, cụ thể ở đây là lớp bệnh "Melanocytic nevi".

akiec	bcc	bk1	df	nv	mel	vasc
1.0	1.0	1.0	1.0	2.5	1.0	1.0

Bảng 3: Tỷ lệ trọng số (class weight) của các phân lớp

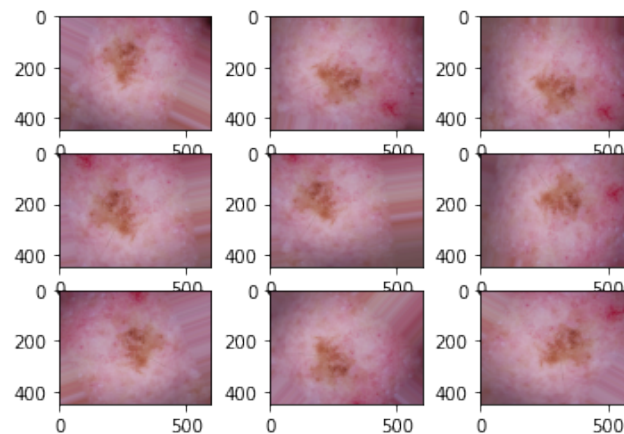
Việc tăng cường dữ liệu (Data Augment) giúp cho tăng dữ liệu dựa trên các dữ liệu đã có mà không cần phải đi thu thập thêm, một số loại dữ liệu đặc thù rất khó thu thập. Và nó cực kỳ hữu ích cho tập dữ liệu HAM10000 khi tập này bị mất cân bằng rất lớn. Để làm giàu tập dữ

liệu bằng việc tăng cường tôi thực hiện các tác động ví dụ như: xoay ảnh, phóng to ảnh, dịch trái, dịch phải ảnh... Và những tham số mà tôi sử dụng được trình bày ở bảng dưới đây.

Tham số	Giá trị	Mô tả
Rotation range	40	Cho phép xoay ảnh từ -40 đến 40 độ.
Width shift range	0.2	Dịch ảnh ngẫu nhiên theo phương ngang với giá trị là 0.2.
Height shift range	0.2	Dịch ảnh ngẫu nhiên theo phương dọc với giá trị là 0.2.
Zoom range	0.2	Phóng to hoặc thu nhỏ ảnh từ giữa ra với giá trị là 0.2.
Horizontal flip	True	Lật ngẫu nhiên ảnh theo phương ngang.
Vertical flip	True	Lật ngẫu nhiên ảnh theo phương dọc.
Fill mode	nearest	Điền các giá trị trống vào các vị trí gần nhau.

Bảng 4: Tham số Data Augment

Với những tham số trên được cấu hình trong hàm **ImageDataGenerator()** của thư viện **Keras** thì ta được kết quả như ảnh dưới đây. Với cách làm này thì dữ liệu sẽ được làm giàu lên rất nhiều và cho hiệu suất tốt hơn là sử dụng một số thuật toán giúp tránh việc mất cân bằng khác như **OverSampling**.



Hình 4: Một bức ảnh khi được làm giàu bằng các tham số trên

2.3 Cấu trúc mạng CNN và huấn luyện mô hình

Nhưng tiêu đề của bài báo, tôi ứng dụng CNN trong việc xây dựng nên model, ảnh dưới đây thể hiện lưu đồ của mà đề tài tài này sẽ thực hiện khi có một bức ảnh hoặc một video (nhiều bức ảnh) khi đưa vào nhận diện.

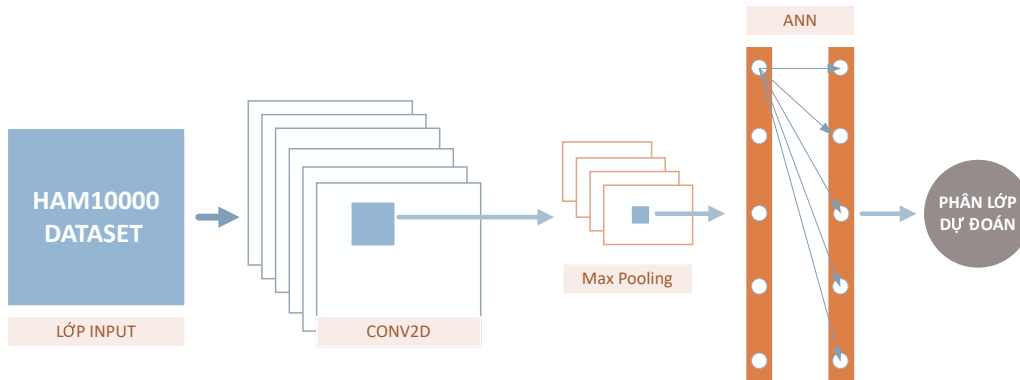


Hình 5: Lưu đồ nhận diện một bức ảnh mới.

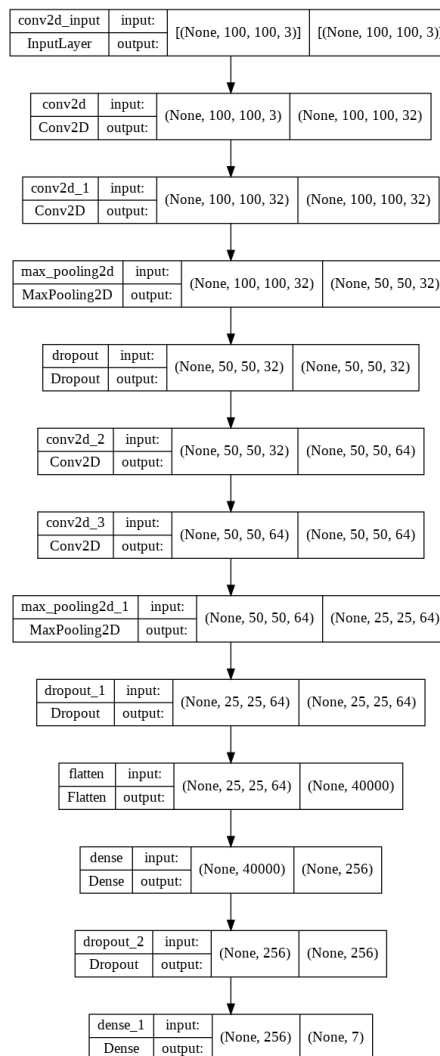
Ở đây bức ảnh hoặc webcam của người dùng sẽ đưa vào mạng gọi chung **Input** - là một bức ảnh, và bức ảnh này sẽ được thay đổi kích thước thành **100x100**, tiếp theo đó Input này sẽ

được đưa vào các lớp trong mạng CNN và cuối cùng sẽ được đi vào các lớp ANN - các lớp được kết nối đầy đủ (Fully connected layer). Xác suất đầu ra cho tất cả các lớp sẽ được tính bằng hàm **Softmax** và lớp có xác suất cao nhất sẽ được hiển thị trên Website.

Hình ảnh dưới đây là sơ đồ mạng Neural Network được tôi xây dựng:



Hình 6: Sơ đồ mạng của đề tài.



Hình 7: Cấu trúc mạng đề tài.

Về huấn luyện (training) mô hình tôi chia tập dữ liệu HAM10000 ra theo tỉ lệ 80:20 với 80% dùng cho training và 20% dùng cho testing và trong đó 15% của dữ liệu training được dùng cho việc Validation. Với kích thước ảnh đầu vào là **100x100** và các nhãn được tôi dùng hàm **to_categorical()** để đưa về dạng **One Hot Encoder** thích hợp trong việc huấn luyện mô hình về phân loại nhiều lớp cụ thể ở đây là 7 lớp.

Với hàm mất mát (**loss function**) tôi lựa chọn sử dụng hàm **Categorical_crossentropy** rất thích hợp khi trước đó tôi đã One Hot Encoder nhãn cùng với đây là bài toán phân loại. Cùng với đó đối với hàm tối ưu (**optimizers**) tôi lựa chọn sử dụng hàm **Adams** với *learning rate* khởi tạo ban đầu là 0.0001.

2.4 Công cụ đánh giá mô hình

Có nhiều cách cũng như nhiều công cụ hay thuật toán dùng để đánh giá hiệu suất cũng như chất lượng của mô hình đó. Ở đề tài này tôi sử dụng những công cụ đánh giá rất phổ biến như: *accuracy, precision, recall, F1-Score*.

Accuracy (Độ chính xác): Độ chính ở đây biểu thị phần trăm lớp bệnh ung thư được phân loại chính xác, được tính bằng công thức dưới đây.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100 \quad (1)$$

Precision: Được hiểu là phần trăm lớp bệnh ung thư dương tính thật (true positive) được nhận diện chính xác.

$$precision = \frac{TP}{TP + FP} \times 100 \quad (2)$$

Recall: Các điểm dương tính thật trong tất cả các dự đoán có thể là dương tính.

$$recall = \frac{TP}{TP + FN} \times 100 \quad (3)$$

F1-Score: Là sự đánh giá dựa trên cả Precision và Recall.

$$F1 - Score = 2 \times \frac{precision \times recall}{precision + recall} \quad (4)$$

2.5 Triển khai mô hình realtime lên website

Triển khai mô hình (model) realtime lên website nhằm tạo ra điều kiện thuận lợi hơn với người sử dụng. Có rất nhiều sự lựa chọn Framework cũng như những ngôn ngữ Backend khác để xây dựng Webserver, với tôi thì do từng trải qua nhiều dự án với Flask Framework nên trong đề tài này tôi cũng sử dụng Flask kết hợp cùng với TensorflowJS một thư viện Javascript của chính chủ Google dành cho những lập trình viên muốn triển khai mô hình Tensorflow lên website.

Cấu trúc Website được tôi thiết kế với 3 chức năng chính: nhận diện realtime sử dụng Webcam hoặc camera, nhận diện thông qua ảnh nội soi tĩnh và chức năng còn lại là trang thông tin về các loại bệnh.

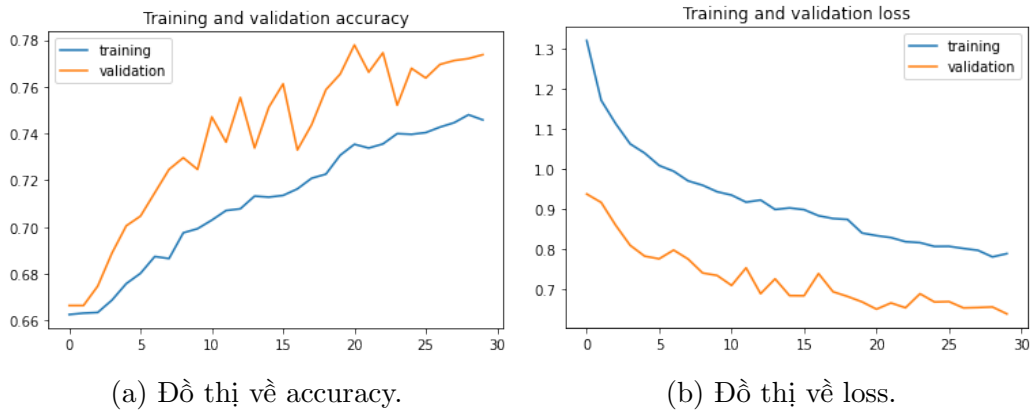
Nói về việc sử dụng dụng TensorflowJS, như trình bày ở phần 2.3 - Input đầu vào là một bức ảnh hoặc một video cụ thể là ảnh được lấy từ webcam hoặc ảnh từ người dùng tải lên. Với bức

ảnh này sẽ được đưa về kích thước 100x100 theo chuẩn Tensor và đưa vào mô hình để nhận diện. Mô hình sẽ trả về xác suất của 7 lớp.

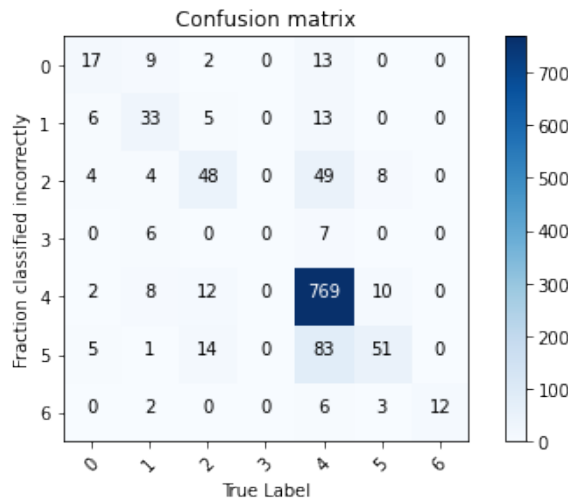
Đọc giả có thể xem tham khảo Source code tôi viết qua QR Code đường dẫn Github mà tôi gắn ở trang đầu bài báo.

3 Kết quả

Từ những phương pháp đã trình bày ở trên trải qua quá trình training và turning mô hình thì tôi đạt được kết quả như sau. Sơ đồ dưới đây là kết quả của quá trình traning.



Hình 8: Kết quả của mô hình.



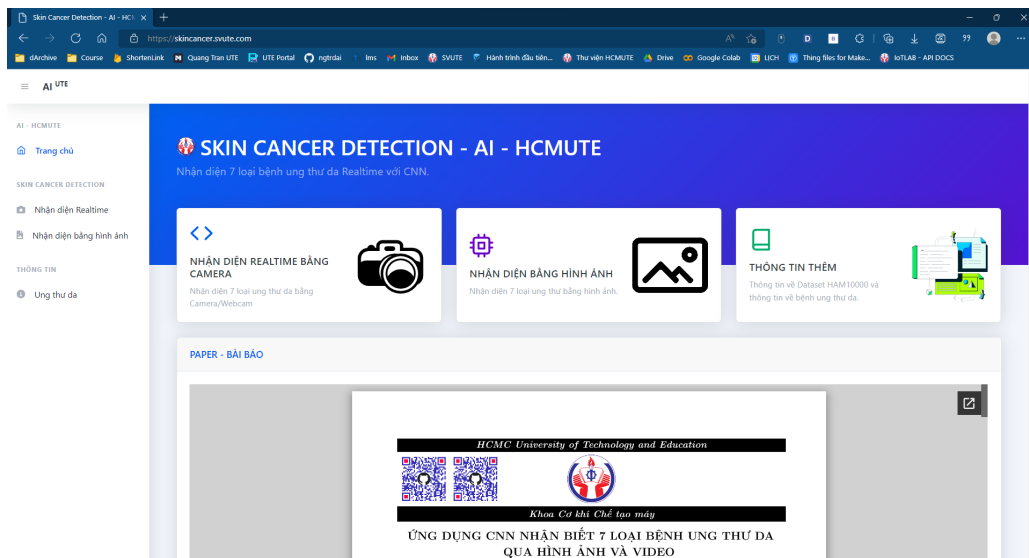
Hình 9: Confusion matrix của mô hình trên tập test.

Kết quả đánh giá mô hình dựa trên những phương pháp đánh giá đã trình bày phần trước ở bản dưới đây.

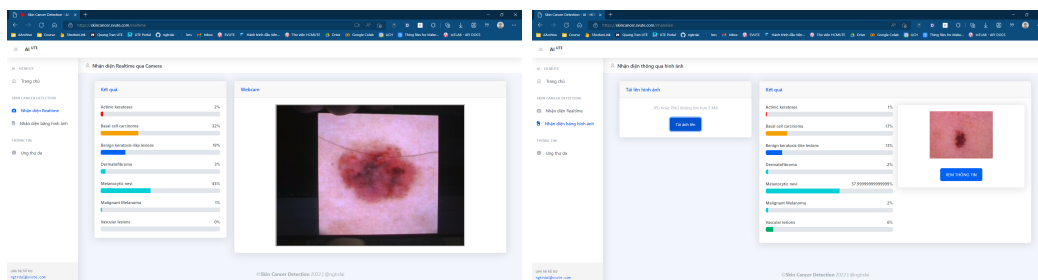
Thuốc đo	Kết quả
Accuracy (%)	77.00%
Precision (%)	59.00%
Recall (%)	46.00%
F1-score (%)	50.00%

Bảng 5: Kết quả đánh giá mô hình.

Sau khi có được mô hình, tôi sử dụng TensorflowJS để chuyển đổi file MODEL.H5 sang file nhị phân để sử dụng trên TensorflowJS. Kết hợp với backend là Flask Framework tôi đã triển khai mô hình lên website. Kết quả đạt được như sau.



Hình 10: Trang chủ website skincancer.svute.com.



(a) Nhận diện realtime qua Camera.

(b) Nhận diện qua hình ảnh.

Hình 11: Kết quả của mô hình.

4 Kết luận

Qua những kết quả trên tôi đưa ra kết luận cho bài báo này như sau, về cơ bản tôi đã xây dựng được mô hình sử dụng các lớp mạng CNN để phân loại mô hình, với tôi là một người chưa có nhiều kinh nghiệm vì thế mạng tôi xây dựng ra chưa đạt kết quả như mong đợi. Nếu có cơ hội được cải tiến đề tài này tôi có thể sử dụng một vài mạng được pretrained để nhằm mục đích trích xuất các đặc trưng của hình ảnh tốt hơn và cho kết quả nhận diện tốt hơn hiện tại. Và bên cạnh đó tôi muốn tìm hiểu thêm về nhiều phương pháp giảm mất cân bằng dữ liệu.

Bên cạnh đó tôi đã triển khai được hệ thống nhận diện lâm sàng trực tuyến thông qua Website tại địa chỉ: <https://skincancer.svute.com> tuy kết quả mô hình chưa được tốt nhưng sản phẩm có thể sử dụng ở mức độ tham khảo.

Lời cảm ơn

Với vai trò là tác giả bài báo này tôi muốn gửi lời cảm ơn đến PGS. TS. Nguyễn Trường Thịnh phụ trách môn học "Trí tuệ nhân tạo" của tôi tại trường đã hỗ trợ kiến thức nền cho tôi để có thể hoàn thành đề tài này. Dù rằng đề tài chưa được hoàn thiện nhưng tôi sẽ tiếp tục phát triển thêm cho đề tài này.

Tài liệu tham khảo

- [1] K SCOTT MADER, "Skin Cancer MNIST: HAM10000 dataset", 2018, [TÀI LIỆU ONLINE]
<https://www.kaggle.com/datasets/kmader/skin-cancer-mnist-ham10000>
 - [2] V.M Thien, L.M Hung, T.K Tam, T.V Lang, "Xử lý mất cân bằng dữ liệu trong phân loại tổn thương da trên ảnh soi da", 2020, DOI: 10.15625/vap.2020.00238
 - [3] Tareq Tayeh, "Skin Lesion Classification using Convolutional Neural Networks", 2020.
 - [4] Nguyen Chien Thang (MIAI), "Có ít dữ liệu, làm sao train model? – Chương 1. Data Augment cho ảnh", 2019. [TÀI LIỆU ONLINE]
<https://www.miai.vn/2019/12/24/co-it-du-lieu-lam-sao-train-model-chuong-1-data-augment/>
-