ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA ĐIỆN – ĐIỆN TỬ
BỘ MÔN VIỄN THÔNG
----------o0o----------

ĐỒ ÁN MÔN HỌC

PHÁT HIỆN ẢNH GIẢ MẠO: MÔ HÌNH CHUYỂN HƯỚNG DỮ LIỆU HIỆU QUẢ VÀ TÍNH TOÁN NHANH

GVHD: **GS.TS. Lê Tiến Thường**
SVTH: **Nguyễn Chính Thụy - 1513372**
Đỗ Tiểu Thiên - 1513172

TP. HỒ CHÍ MINH, THÁNG 6 NĂM 2018

ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA

----- ☆ -----

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc.

----- ☆ -----

Khoa: **Điện – Điện tử**
Bộ Môn: **Viễn thông**

NHIỆM VỤ ĐỒ ÁN MÔN HỌC

1. HỌ VÀ TÊN: **NGUYỄN CHÍNH THỤY** MSSV: 1513372
   ĐỖ TIỂU THIÊN 1513172

2. NGÀNH: **ĐIỆN TỬ - VIỄN THÔNG** LỚP : DD15KSVT
3. Đề tài: **PHÁT HIỆN ẢNH GIẢ MẠO: MÔ HÌNH CHUYỂN HƯỚNG DỮ LIỆU HIỆU QUẢ VÀ TÍNH TOÁN NHANH**
4. Nhiệm vụ (Yêu cầu về nội dung và số liệu ban đầu):
   - Khảo sát các phương pháp Phát hiện Ảnh giả mạo, và đề xuất một phương pháp Phát hiện Ảnh giả mạo.
   - Triển khai phương pháp đề xuất bằng phần mềm.
   - Tiến hành thí nghiệm phương pháp đề xuất trên một tập cơ sở dữ liệu chuẩn, để đánh giá độ chính của mô hình.
   - Độ chính xác của mô hình phải lớn hơn 80% trên tập cơ sở dữ liệu chuẩn dùng trong phần thí nghiệm.
   - Đánh giá ưu khuyết điểm của phương pháp, cũng như đề xuất hướng nghiên cứu trong tương lai.
5. Ngày giao nhiệm vụ đồ án: 01/04/2018
6. Ngày hoàn thành nhiệm vụ: 01/06/2018
7. Họ và tên người hướng dẫn: Phần hướng dẫn
   GS.TS. Lê Tiến Thường Tất cả các nhiệm vụ nêu trên

Nội dung và yêu cầu ĐAMH đã được thông qua Bộ Môn.

Tp.HCM, ngày..... tháng..... năm 2018

CHỦ NHIỆM BỘ MÔN **NGƯỜI HƯỚNG DẪN CHÍNH**

PHẦN DÀNH CHO KHOA, BỘ MÔN:
Người duyệt (chấm sơ bộ):.....
Đơn vị:.....
Ngày bảo vệ : .....
Điểm tổng kết: .....
Nơi lưu trữ luận văn: .....

LỜI CẢM ƠN

Trong quá trình thực hiện đồ án này, nhóm đã nhận được sự hỗ trợ nhiệt thành, sự hướng dẫn tận tình, và những lời động viên tích cực từ gia đình, các thầy cô ở khoa Điện – Điện tử, và bạn bè. Nhờ vào đó, nhóm đã hoàn thành tốt đồ án như mong đợi ban đầu.

Lời đầu tiên, nhóm xin được gửi lời cảm ơn chân thành đến thầy hướng dẫn GS.TS. Lê Tiến Thường, người đã trực tiếp hướng dẫn nhóm thực hiện nghiên cứu từ đầu đồ án cho đến khi hoàn thành. Trong khoảng thời gian thực hiện, cũng như viết báo cáo cho đồ án, nhờ vào những định hướng, nhận xét, và góp ý của thầy, nhóm đã có thể hoàn thành tốt những mục tiêu đã đề ra khi nhận đồ án.

Tiếp theo, nhóm cũng muốn được gửi những lời cảm ơn đến các thầy cô tại trường đại học Bách Khoa Tp.HCM vì đã truyền đạt những kiến thức giá trị trong khoảng thời gian nhóm theo học tại đây. Ngoài kiến thức, các thầy cô còn chia sẻ những kinh nghiệm quý báu và đưa những lời khuyên xác đáng khi nhóm cần định hướng. Tất cả những điều đó đã và đang giúp nhóm rất nhiều trên con đường chinh phục tri thức ở môi trường học thuật cũng như là môi trường làm việc sau này.

Bên cạnh đó, nhóm xin được gửi lời cảm ơn đến những lời động viên khuyến khích từ gia đình, đặc biệt là ba mẹ, trong khoảng thời gian theo học tại trường đại học Bách Khoa Tp.HCM.

Cuối cùng, nhóm cũng muốn gửi lời cảm ơn đến tất cả bạn bè, đặc biệt là tập thể lớp Kĩ sư tài năng Điện tử - Viễn thông, đã cũng đồng hành với nhóm trong 2 năm học vừa qua.

Tp. Hồ Chí Minh, ngày 01 tháng 6 năm 2018.

Sinh viên

TÓM TẮT ĐỒ ÁN

Ở đồ án này, nhóm tìm hiểu các hướng nghiên cứu trong lĩnh vực ***Nhận diện Ảnh giả mạo (Image Forgery Detection)***. Sau khi tìm hiểu, đánh giá ưu khuyết điểm các hướng đi, nhóm quyết định chọn cách ***tiếp cận dữ liệu (Data-driven)*** để giải quyết bài toán Phát hiện ảnh giả mạo. Cụ thể, bài toán Phát hiện ảnh giả mạo là chỉ ra một bức ảnh là giả mạo hay không, chứ không chỉ ra những vùng nào trong bức ảnh bị giả mạo.

Về cách tiếp cận Dữ liệu, nhóm đề xuất một ***mạng Nơ-ron nhân tạo (Artificial Neural Network)*** để tự động học các đặc trưng ảnh, nhằm phân loại ảnh giả và ảnh thật. Cụ thể, bức ảnh cần kiểm tra sẽ được duyệt bằng ***cửa sổ dịch (Sliding window)***, mỗi cửa sổ dịch tương ứng với một ***khối ảnh (patch)***. Sau đó, mỗi khối ảnh được trích xuất đặc trưng bằng biến đổi Wavelets, và được đẩy vào mạng Nơ-ron. Sau khi mạng Nơ-ron ***gán nhãn (label)*** cho các khối ảnh, một tầng ***Hậu xử lý (post-processing)*** được sử dụng để lọc bỏ các nhãn có độ tin cậy thấp. Cuối cùng, một ảnh được coi là không bị giả mạo nếu tất cả các nhãn đều ***âm tính (negative)***. Ngược lại, nó sẽ bị coi là giải mạo nếu có ít nhất một nhãn ***dương tính (positive)***.

Trong quá trình nghiên cứu, nhóm đã phát hiện ra các đặc trưng của kênh màu Y trong YCrCb không đóng góp nhiều vào việc nhận dạng. Do đó, bằng cách lược bỏ đi những đặc trưng của kênh Y, độ dài vector đặc trưng chỉ còn lại 2/3 so với ban đầu. Kết quả thực nghiệm cho thấy, thời gian ***huấn luyện (training)*** cũng như thời gian chạy ***thực tế (testing)*** đã được giảm đi trong khi độ chính xác lại tốt hơn so với khi chưa giảm chiều dữ liệu. Ngoài ra, mô hình của nhóm cũng khá ***cạn (narrow)*** khi so sánh với các phương pháp sử dụng ***Học sâu (Deep Learning)***, nhưng độ chính xác vẫn không thua kém. Điều này sẽ cực kì thích hợp khi triển khai ở đa số các máy tính sinh viên Việt Nam có cấu hình không được mạnh.

MỤC LỤC

DANH SÁCH HÌNH MINH HỌA

DANH SÁCH BẢNG SỐ LIỆU

DANH SÁCH THUẬT NGỮ VIẾT TẮT

Thuật ngữ tiếng Anh	
Thuật ngữ viết tắt	Thuật ngữ đầy đủ
BMP	Bitmap
CNN	Convolutional Neural Network
CWT	Continuous Wavelets Transform
DCT	Discrete Cosine Transform
DWT	Discrete Wavelets Transform
FCNN	Fully Connected Neural Network
JPEG	Joint Photographic Experts Group
LSTM	Long Short Term Memory
ReLU	Rectifier Linear Unit
RGB	Red – Green – Blue
RNN	Recurrent Neural Network
SIFT	Scale-Invariant Feature Transform
SRM	Spacial Rich Model
SURF	Speeded Up Robust Features
SVM	Support Vector Machine
TIFF	Tagged Image File Format

Chương 1: GIỚI THIỆU ĐỀ TÀI

1.1 Tổng quan về Phát hiện ảnh giả mạo

Trong những năm gần đây, khi mà công nghệ số phát triển mạnh mẽ, thì ***thông tin đa phương tiện (multimedia information)*** cũng đang trở nên rất phổ biến trong các hầu hết mọi khía cạnh sống và sinh hoạt của con người. Một trong những loại thông tin được sử dụng nhiều nhất đó là hình ảnh. Ta có thể thấy hình ảnh số trong rất nhiều lĩnh vực như giáo dục, y tế, truyền thông, giải trí, quốc phòng, an ninh, khoa học, kỹ thuật, nghệ thuật, vv. Chính sự ra đời của các máy ảnh nhỏ gọn, cũng như máy ảnh được tích hợp vào điện thoại thông minh đã tạo ra sự thịnh hành của ảnh số, bất kì ai cũng có thể chụp ảnh để lưu giữ khoảnh khắc.

Để phục vụ cho các nhu cầu thẩm mĩ về ảnh, đã có rất nhiều phần mềm chỉnh sửa ảnh được thiết kế (như Adobe Photoshop, Paint.NET, PhotoScape) nhằm cung cấp cho người sử dụng một bộ công cụ mạnh mẽ trong việc chỉnh sửa ảnh theo ý muốn. Điều này đã đem lại rất nhiều lợi ích cho cộng đồng sử dụng ảnh, vì họ có thể biến đổi một bức ảnh gốc thành ảnh có màu, họa tiết, vật thể đẹp hơn. Tuy nhiên bên cạnh những lợi ích mà nó mang đến, những công cụ chỉnh sửa ảnh này có thể gây ra những ảnh hưởng tiêu cực. Ở một vài quốc gia, bao gồm cả Việt Nam, luật pháp cho phép sử dụng ảnh như là một tài liệu để làm bằng chứng. Tuy nhiên, nếu bức ảnh đã qua chỉnh sửa bằng phần mềm có thể sẽ làm thay đổi nội dung thật của bức ảnh, từ đó làm cho bằng chứng sai lệch, ảnh hưởng đến kết luận của tòa án. Ngoài ra, còn rất nhiều trường hợp đòi hỏi tấm ảnh phải phản ánh đúng sự thật, hay nói cách khác, ảnh phải là ảnh gốc, chưa qua chỉnh sửa. Chính vì những nhu cầu thực tiễn này, đã dẫn tới sự ra đời của ngành Giám định ảnh số.

Nhiệm vụ của ngành Giám định ảnh chính là đi kiểm tra một tấm ảnh, liệu rằng nó có phải là ảnh gốc hay không, từ đó đưa ra độ tin cậy cho bức ảnh đó. Do hầu như ảnh ở dạng số, nên ở hiện tại, ngành Giám định ảnh tập trung vào ảnh số. Các giải thuật sẽ được thiết kế và chạy trên máy tính để kiểm tra ảnh. Có nhiều cách để phân nhóm giải thuật, trong đó, dựa vào cách thức chỉnh sửa ảnh mà ta có thể chia các giải thuật thành hai hướng đó là ***Phát hiện Sao chép-Dịch chuyển (Copy-Move Detection)***, và ***Phát hiện Nối ảnh (Splicing Detection)***.

Đối với loại đầu tiên (***Sao chép-Dịch chuyển***), một phần của bức ảnh sẽ được sao chép và dán lại vào một vị trí khác ngay trong bức ảnh đó. Đây là một cách chỉnh sửa ảnh thường gặp nhất, vì tính dễ thực hiện của nó. Còn đối với loại thứ hai (***Nối ảnh***), từ một bức ảnh, người chỉnh sửa thực hiện sao chép vùng ảnh, và dán lại vào một ảnh khác. Cách chỉnh sửa này sẽ khó phát hiện hơn so với cách đầu tiên đã nêu ở trên. Điều này có thể được giải thích rằng đối với Sao chép và dịch chuyển, ta hoàn toàn có thể tìm kiếm thông tin trong bức ảnh để làm cơ sở và tìm ra vùng sao chép. Tuy nhiên, ở loại thứ hai, ta không thể tìm được bất kì thông tin nào để làm cơ sở đối chiếu. Chính vì vậy, có nhiều thách thức kĩ thuật đặt ra hơn với vấn đề giả mạo bằng phương pháp nối ảnh.

1.2 Nghiên cứu trong và ngoài nước

1.2.1 Phát hiện Sao chép – Dịch chuyển

Có rất nhiều phương pháp được đề xuất để giải quyết vấn đề phát hiện hình ảnh giả mạo, tuy nhiên, phát hiện sao chép – dịch chuyển là một trong những cách tiếp cận phổ biến nhất vì tính đặc trưng của cách tạo hình ảnh giả mạo. Hướng tiếp cận này chia thành hai nhóm: Key-point-based và Block-based.

Đầu tiên, [1] [2] [3] trước đây thường trích xuất các tính năng của điểm khóa trong hình ảnh, dựa trên kỹ thuật SIFT nổi tiếng và SURF. Sau đó, các đặc trưng của các điểm khóa được so sánh để tìm điểm tương tự. Các vùng giả cuối cùng được chỉ ra khi quá trình so sánh hoàn thành và vượt qua một mức ngưỡng. Phương pháp này có tác dụng cho việc phát hiện biến dạng hình học và trùng lặp. Tuy nhiên, trong trường hợp đối tượng trùng lặp chứa ít cấu trúc thì hai kỹ thuật này không hiệu quả, dẫn đến suy giảm hiệu suất của thuật toán.

Trong phương pháp thứ hai [4] [5] [6] [7], đặc trưng của các khối, được tạo ra bởi cửa sổ trượt. Sau đó, các đặc trưng được đưa vào quá trình so sánh và khối ấy xem là vùng trùng lặp khi quá trình so sánh có sự tương đồng đủ lớn.

Mặc dù phương pháp phát hiện sao chép – dịch chuyển này được áp dụng rộng rãi nhưng nhược điểm của phương pháp là không thể phát hiện vùng ảnh giả được ghép từ một ảnh khác.

1.2.2 Phương pháp dựa vào định dạng ảnh JPEG

Bởi vì hầu như ảnh được định dạng JPEG nên có nhiều nghiên cứu dựa trên định dạng này để phát hiện giả mạo. Nếu có ai đó chỉnh sửa trên ảnh gốc JPEG thì nó sẽ xảy ra sự nén kép JPEG, đó là bằng chứng cho các nhà khoa học tìm tìm dấu vết để phát hiện ảnh giả mạo.

Bằng nghiên cứu biến đổi Discrete Cosine Transformation (DCT), [8] thiết kế một phương pháp để nhận diện ảnh giả, dựa trên lượng tử hóa kép DCT. Ưu điểm của phương pháp này là nhanh. Hơn nữa, đây là phương pháp đầu tiên tự động xác định vị trí vùng bị làm giả. Wang et al. [9] cũng sử dụng tính chất của biến đổi DCT để xử lí phát hiện ảnh giả. Theo giả thuyết các phân phối khác nhau của các vùng giả, họ tính xác xuất các khối DCT giả mạo. Bên cạnh đó, họ thiết kế 3 loại đặc trưng để phân biệt các mẫu thực sự là dương tính từ những mẫu dương tính giả. Để phát hiện sự nén lại trong JPEG, tác giả trong [10] đề xuất một mô hình đặc

điểm tuần hoàn trong miền không gian và miền DCT. Phương pháp này có thể xử lí cho cả aligned và non- aligned nén kép JPEG. Thing et al. [11] giới thiệu một phương pháp phát hiện tuần hoàn của lượng tử kép. Rõ ràng, họ khai thác tính chất của phân bố Gauss của DCT histograms của ảnh tự nhiên. Bằng cách khám phá JPEG ghost, [12] giới thiệu một phương pháp tự động phát hiện các vùng nén đơn và nén đôi. Phương pháp này mạnh mẽ cho cả aligned và shifted JPEG.

Trong [13], Bianchi et al. đề xuất phương pháp Bayesian để tự động tính bản đồ xác xuất nén kép của mảnh 8x8 trong bức ảnh. Giả sử một ảnh giả mạo đại diện cho một phép nén kép, nó đòi hỏi phải xác minh giả định này trước khi tiến hành thuật toán phát hiện ảnh giả. Tuy nhiên, phương pháp này không cần phải kiểm tra vùng đáng ngờ cho dù nó có nén kép.

Trong [14], Chang et al. phát hiện một thuật toán mới phát hiện ảnh giả mạo. Phương pháp này có hai giai đoạn, đầu tiên phát hiện các vùng nghi ngờ và sau đó áp dụng phương pháp mới là Multi-Region Relation để phát hiện vùng giả mạo từ các vùng ở giai đoạn trước. ưu điểm của phương pháp này là tốc độ nhanh và có thể nhận biết được ảnh bao gồm nền đồng nhất.

Tóm lại, cách giải quyết này là hiệu quả trong trường hợp ảnh là JPEG và nén kép. Tuy nhiên, trong các định dạng ảnh khác thì phương pháp này không hoạt động tốt. Vì vậy, torng thực tế thì phương pháp này khó để áp dụng.

1.2.3 Phương pháp chuyển hướng dữ liệu (data-driven)

Các phương pháp hướng dữ liệu hiện ngày càng được sử dụng rộng rãi trong lĩnh vực paht1 hiện ảnh giả mạo. Trong [15], một mạng Nơ-ron xoắn (CNN) được sử dụng để để lọc trung vị. Sau đó, Bayar et al. [16] giới thiệu một lớp xoắn trong mô hình CNN của họ để giải quyết phát hiện dấu vết. Thúc đẩy bởi các ứng dụng của CNN, Rao et al. [17] đề xuất một mô hình CNN và lớp đầu tiên của mạng được khởi tạo SRM để giảm nội dung hình ảnh cũng như hiện vật dự trữ. Một phương pháp khác, Ouyang et al. [18] đã sử dụng hướng tiếp cận transfer-learning để giải quyết vấn đề phát hiện sao chép - dịch chuyển bằng cách sử dụng kiến trúc được huấn luyện trước của AlexNet trong [22]. Bên cạnh đó, [19] trích xuất các đặc trưng của các mảnh giả trong vật thể giả mạo bằng biến đổi Daubechies Wavelets và sau đó đưa chúng vào bộ Stacked Auto-Encoder để phân loại.

Được truyền cảm hứng cho các phương pháp chuyển hướng dữ liệu mới nổi, nhóm đề xuất một mô hình chuyển hướng dữ liệu để giải quyết vấn đề phát hiện ảnh giả mạo. Mô hình có thể làm rõ trong nhiều ngữ cảnh mà các mô hình khác thông thường không thể giải quyết được. Đặc biệt, nhóm sử dụng phương pháp trích xuất đặc trưng trong [19] và tiến hành xem xét để phân tích hiệu năng của việc trích xuất đặc trưng. Tiếp theo, nhóm thiết kế một mạng Nơ-ron nhân tạo để phân loại các tính năng được trích xuất, đi kèm với một phương pháp lựa chọn mẫu để giúp mạng tìm hiểu sự khác biệt giữa ảnh thật và giả.

1.3 Nhiệm vụ

- **Yêu cầu:**
  + Nội dung 1: Tìm hiểu tổng quan về các phương pháp giám định ảnh.
  + Nội dung 2: Đánh giá các ưu khuyết điểm của các hướng nghiên cứu, cũng như là của các phương pháp trong những hướng nghiên cứu đó.
  + Nội dung 3: Lựa chọn một hướng nghiên cứu để tìm hiểu sâu vào.
  + Nội dung 4: Đề xuất một phương án phát hiện ảnh giả mạo.
  + Nội dung 5: Tiến hành thí nghiệm đề xuất để kiểm chứng.

- **Kết quả cần đạt:**
  + Nội dung 1: Đề xuất một giải thuật phát hiện ảnh giả mạo.
  + Nội dung 2: Tiến hành kiểm tra giải thuật trên một tập cơ sở dữ liệu chuẩn, để đưa ra các thông số đánh giá chất lượng của giải thuật.
  + Nội dung 3: Dùng các thông số đánh giá chất lượng của giải thuật mà sinh viên đề xuất, so sánh với các giải thuật của các nghiên cứu cùng lĩnh vực, trong và ngoài nước.

- **Giới hạn đề tài:**
  + Nội dung 1: Giải thuật có thể triển khai được trên các máy tính để bàn và máy tính xách tay thông dụng.
  + Nội dung 2: Các thông số đánh giá chất lượng của giải thuật là dựa trên một tập cơ sở dữ liệu chuẩn cho Giám định ảnh số, không phải là ảnh ngẫu nhiên.
  + Nội dung 3: Giải thuật kiểm tra ảnh phải chỉ ra được ảnh là ảnh giả hay ảnh thật, không cần chỉ ra những vị trí bị chỉnh sửa trong bức ảnh.
  + Nội dung 4: Thông số độ chính xác (accuracy) của giải thuật phải lớn hơn ngưỡng 80%.

Chương 2: LÝ THUYẾT WAVELETS VÀ MẠNG NƠ-RON

2.1 Biến đổi Wavelets

Phép biến đổi Wavelets liên tục của một hàm số $f(t)$ có nguồn gốc từ một hàm Wavelets mẹ $\psi(t)$. Hàm Wavelets mẹ này có thể là một hàm thực liên tục hoặc phức liên tục, và phải thỏa mãn 2 tính chất sau đây:

- Tích phân trên toàn miền thời gian của hàm Wavelets mẹ bằng 0:

$$\int_{-\infty}^{+\infty} \psi(t)dt = 0 \tag{2.1}$$

- Năng lượng của hàm Wavelets mẹ là một số hữu hạn:

$$\int_{-\infty}^{+\infty} |\psi(t)|^2 dt < \infty \tag{2.2}$$

Sau khi lựa chọn $\psi(t)$, phép biến đổi Wavelets liên tục của một hàm khả tích bình phương $f(t)$ được tính bằng công thức:

$$W(a,b) = \int_{-\infty}^{+\infty} f(t) \frac{1}{\sqrt{|a|}} \psi^* \left(\frac{t-b}{a}\right) dt \tag{2.3}$$

Phép biến đổi này là một hàm số của hai thông số thực a và b. Nếu ta muốn định nghĩa một hàm $\psi_{a,b}(t)$, thì (2.3) sẽ trở thành dạng rút gọn như bên dưới:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{|a|}} \psi^* \left(\frac{t-b}{a}\right) \tag{2.4}$$

$$W(a,b) = \int_{-\infty}^{+\infty} f(t)\psi_{a,b}(t)dt \tag{2.5}$$

Một cách toán học, phương trình (2.5) được gọi là tích vô hướng của $f(t)$ và $\psi_{a,b}(t)$. Trong đó, $\frac{1}{\sqrt{|a|}}$ là hệ số chuẩn hóa để đảm bảo năng lượng của $\psi_{a,b}(t)$ không phụ thuộc vào a và b.

$$\int_{-\infty}^{+\infty} |\psi_{a,b}(t)|^2 dt = \int_{-\infty}^{+\infty} |\psi(t)|^2 dt \tag{2.6}$$

Với mỗi giá trị của a, $\psi_{a,b}(t)$ là bản sao của $\psi_{0,b}(t)$, mà ở đó thời gian bị dịch đi b đơn vị. Do đó, b được gọi là thông số dịch. Khi đặt $b = 0$, ta có:

$$\psi_{a,0}(t) = \frac{1}{\sqrt{|a|}}\psi\left(\frac{t}{a}\right) \tag{2.7}$$

Phương trình (2.7) cho thấy a là thông số co dãn. Khi $a > 1$, hàm Wavelets sẽ được dãn ra, trong khi nếu $0 < a < 1$, hàm sẽ thu lại. Sau đó, biến đổi nghịch của CWT được đinh nghĩa. Đặt $\Psi(\omega)$ là biến đổi Fourier của $\psi(t)$:

$$\Psi(\omega) = \int_{-\infty}^{+\infty} \psi(t)e^{-j\omega t}dt \tag{2.8}$$

Nếu $W(a,b)$ là biến đổi CWT của $f(t)$ sử dụng hàm Wavelets $\psi(t)$, biến đổi nghịc CWT sẽ được tính theo công thức:

$$f(t) = \frac{1}{C}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \frac{1}{|a|^2}W(a,b)\psi_{a,b}(t)dadb \tag{2.9}$$

Trong đó, C được định nghĩa như sau:

$$C = \int_{-\infty}^{+\infty} \frac{|\Psi(\omega)|^2}{|\omega|}d\omega \tag{2.10}$$

Biến đổi CWT chỉ tồn tại nếu C là số dương hữu hạn. Do đó, C là điều kiện tồn tại của biến đổi Wavelets. Cùng với 2 điều kiện được đề cập ở trên, đây là điều kiện thứ ba mà một hàm số cần phải thỏa mãn để trở thành một hàm Wavelets. Chúng ta có thể xem xét CWT như một ma trận hai chiều của tích vô hướng giữa $f(t)$ và $\psi_{a,b}(t)$. Các dòng của ma trận tương ứng với các giá trị của a, còn các cột thì tương ứng với các giá trị của b.

$$\langle f(t), g(t)\rangle = \int_{-\infty}^{+\infty} f(t)g^*(t)dt => \langle f(t), \psi_{a,b}(t)\rangle$$

$$= \int_{-\infty}^{+\infty} f(t)\psi_{a,b}(t)dt \tag{2.11}$$

Việc tính toán tất cả các hệ số Wavelets là rất phức tạp. Bên cạnh đó, điều này cũng chiếm một lượng lớn dung lượng bộ nhớ. Để giảm thiểu độ số lượng các phép tính, chúng ta chỉ lựa chọn một nhóm nhỏ các tỉ lệ và vị trí để tính toán. Hơn thế nữa, nếu như ta chuyển bài

toán về để xử lý dưới dạng nhị phân thì kết quả đạt được sẽ nhanh hơn và chính xác hơn. Việc chọn lựa các tỉ lệ và vị trị để tính toán ở trên sẽ tạo ra một ***lưới điện tử (dyadic grid)***. Từ đó chúng ta thấy rằng việc tính toán DWT thực chất là một dạng số hóa CWT. Việc số hóa này sử dụng 2 hệ số a, b ở bên dưới:

$$a = 2^m ; b = 2^m n ; (m, n \in Z) \tag{2.12}$$

Hướng tiếp cận toán học của DWT dựa trên sự thật rằng hàm số $f(t)$ có thể được biểu diễn bằng các hàm tuyến tính cơ bản:

$$f(t) = \sum_k a_k \psi_k(t) \tag{2.13}$$

Trong đó, a_k là hệ số phân tích, và ψ_k là hàm phân tích hay là hàm cơ bản. Nếu các hàm cơ bản trực giao với nhau, các hệ số có thể được xấp xỉ bằng công thức sau:

$$a_k = \langle f(t), \psi_k(t) \rangle = \int f(t)\psi_k(t)dt \tag{2.14}$$

Trong đó, $f(t)$ được tính từ phương trình (2.13). Ví dụ, các hàm cơ bản trực giao của biến đổi Fourier là $\sin(kw_0 t)$ và $\cos(kw_0 t)$.

Đối với DWT, ta có một vài họ Wavelets cơ bản. Đầu tiên là họ **Haar**. Biến đổi Wavelets Haar là biến đổi cơ bản và đơn giản nhất trong tất cả các họ Wavelets. Hình ***Figure 1*** cho thấy hình dạng của hàm $\psi(t)$ biến đổi Haar. Bởi vì tính đơn giản của nó, biến đổi Haar được áp dụng rất nhiều trong nén ảnh.

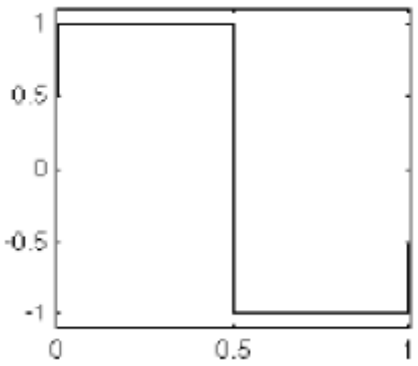


Figure 1 – Hàm $\psi(t)$ trong biến đổi Haar

Bên cạnh đó, **Yves Meyer** cũng là một trong các họ phổ biến của biến đổi DWT. Biến đổi này có khả năng phân tích tín hiệu tốt hơn biến đổi Haar. Hình dạng của hàm $\psi(t)$ trong biến đổi Meyer như ở hình ***Figure 2***.

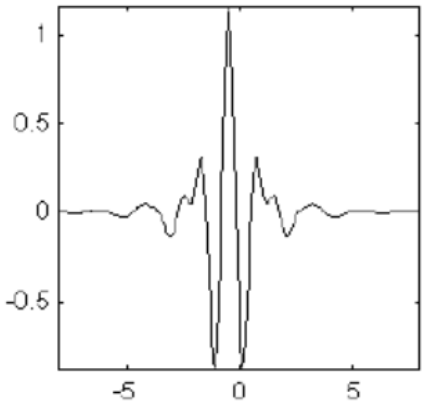


Figure 2 – Hàm $\psi(t)$ trong biến đổi Meyer

Ngoài ra, Daubechies là một họ mạnh mẽ và cũng chính là một trong những họ phức tạp nhất trong DWT. Biến đổi này được áp dụng rộng rãi như trong chuẩn nén JPEG−2000. Hình ***Figure 3*** cho thấy một vài hàm $\psi(t)$ của họ Daubechies.

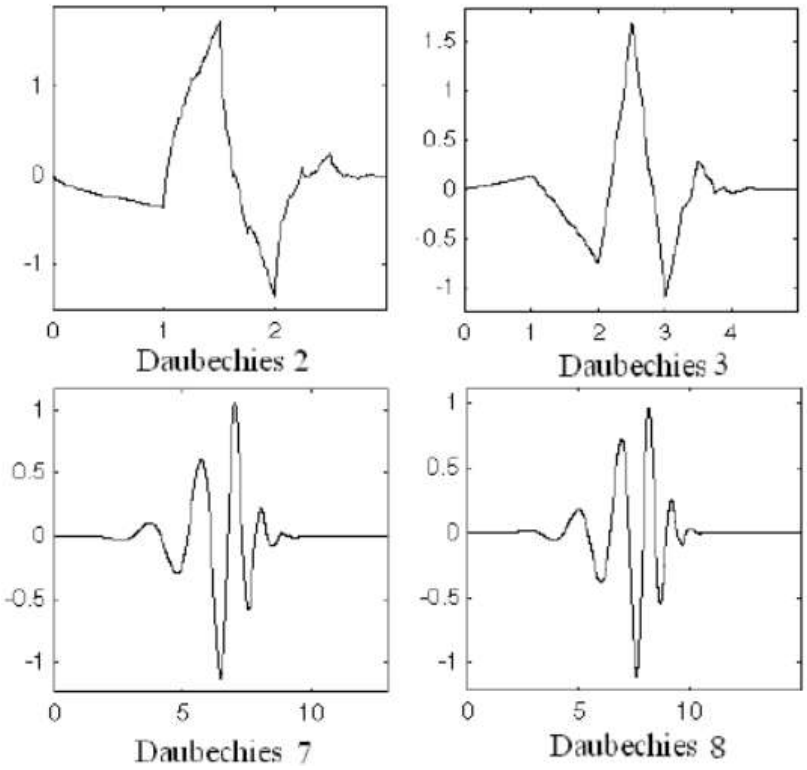


Figure 3 - Hàm ψ(t) của biến đổi Daubechies với bậc n=2,3,7,8

2.2 Mạng Nơ-ron nhân tạo

Trong bài toán phân loại, mạng Nơ-ron nhân tạo là một phương pháp phổ biến và mạnh mẽ. Trước khi mạng Nơ-ron nhân tạo được ứng dụng phổ biến, người ta thường sử dụng các phương pháp phân loại tuyến tính như Logistic và Softmax. Tuy nhiên, vấn đề gặp phải là các phương pháp này chỉ phân loại được dữ liệu tuyến tính hoặc gần tuyến tính. Nói cách khác, giả sử dữ liệu được biểu diễn trong một miền không gian, các phương pháp tuyến tính nêu trên chỉ có thể phân loại hiệu quả nếu ta tìm được các siêu phẳng để chia dữ liệu ra thành các phần riêng biệt với nhau. Chính vì vậy, các phương pháp này không thể dùng để phân loại cho dữ liệu phi tuyến.

Ví dụ, ở ***Figure 4***, dữ liệu được biểu diễn bằng những điểm trong không gian 2 chiều. Ta thấy rằng các điểm trong mỗi lớp nằm thành cụm và tách biệt khỏi nhau. Nhờ đó, ta có thể tìm được hai đường thẳng để chia tất cả các điểm trên thành 3 loại (đỏ, lục, lam). Đây là trường hợp mà ta có thể sử dụng các phương pháp tuyến tính để phân loại. Tuy nhiên, ở ***Figure 5***, ta không thể nào tìm được một đường thẳng để làm ranh giới cho các phân lớp. Trong trường hợp này, ta không thể dùng mô hình tuyến tính để phân loại.

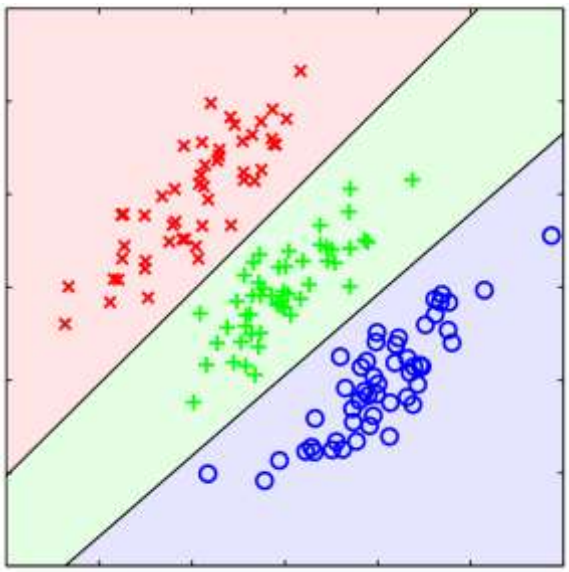


Figure 4 - Dữ liệu tuyến tính

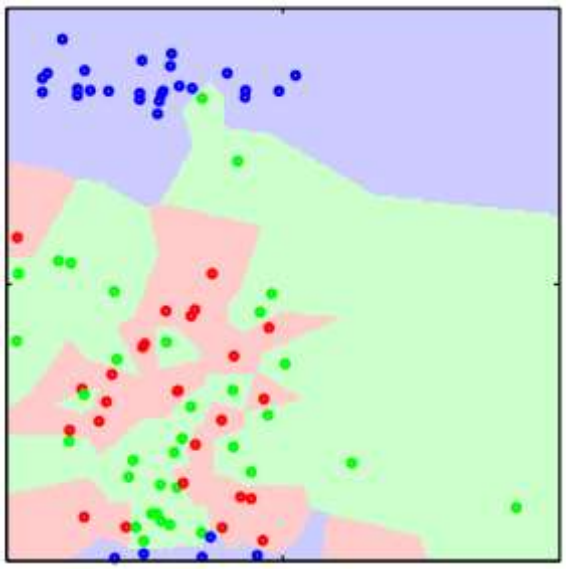


Figure 5 - Dữ liệu không tuyến tính

Do vậy, để có thể phân loại được những kiểu dữ liệu phi tuyến, phương pháp sử dụng, tự bản thân nó, phải chứa các phần tử phi tuyến. Và mạng Nơ-ron nhân tạo đáp ứng được điều này. Cụ thể, một mạng Nơ-ron nhân tạo bao gồm nhiều ***lớp (layer)***, mỗi lớp gồm các ***nơ-ron***

(neural). Hình ***Figure 6*** mô tả mô hình của một mạng nơ-ron nhận tạo, ở đó, các nơ-ron của lớp này được kết nối với tất cả các nơ-ron trong lớp ngay sau nó (mô hình FCNN).

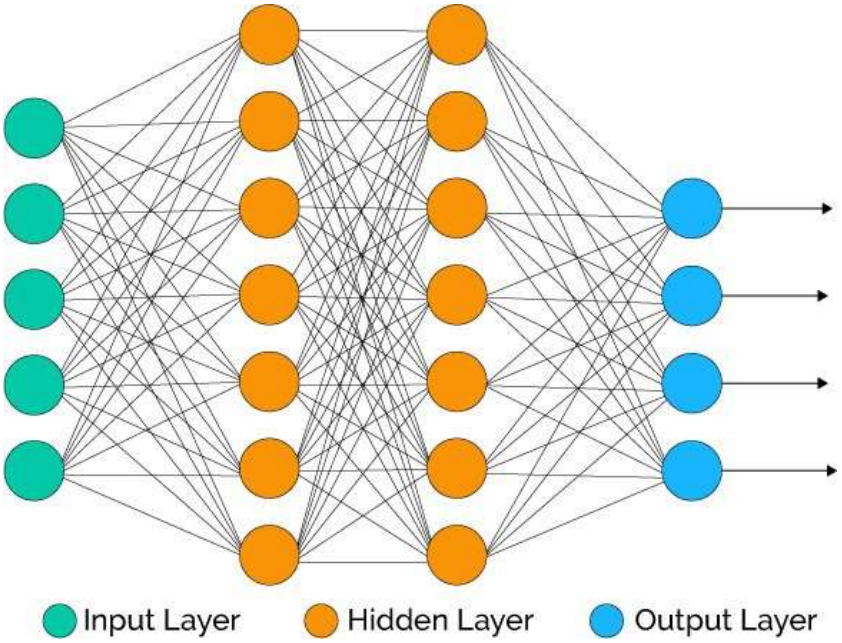


Figure 6 - Mạng nơ-ron nhân tạo

Một cách toán học, tại lớp l, giá trị của các nơ-ron nằm trong nó sẽ được tính dựa vào giá trị của các nơ-ron trong lớp ngay trước nó.

$$a^{(l)} = f\left(W^{(l)}a^{(l-1)} + b^{(l)}\right)$$

Trong đó, $W^{(l)}$ và $b^{(l)}$ là ***trọng số (weight)*** và ***hệ số tự do (bias)*** trong liên kết giữa lớp l và $l-1$, $a^{(l)}$ là giá trị của các nơ-ron trong lớp l, và $f(x)$ là ***hàm số kích hoạt (activation function)*** của lớp đó. Hàm số kích hoạt này thường là những hàm phi tuyến, như ***sigmoid***, ***tanh***, và ***ReLU***. Nhờ đó, mà mạng nơ-ron nhân tạo khác biệt với các mô hình tuyến tính truyền thống và có khả năng để giải quyết những bài toán dữ liệu phi tuyến.

Ngoài ra, mạng nơ-ron nhân tạo còn có nhiều biến thể, để thích hợp với từng bài toán cụ thể. Mạng ở ***Figure 6*** còn được gọi là ***mạng nơ-ron kết nối đủ (Fully-connected Neural Network).*** Đối với các bài toán ảnh, ***mạng nơ-ron xoắn (Convolutional Neural Network)*** thường được dùng, trong khi ***mạng nơ-ron hồi quy (Recurrent Neural Network)*** lại được dùng trong ***xử lý ngôn ngữ tự nhiên (Nature Language Processing).***

Chương 3: ĐỀ XUẤT VÀ TRIỂN KHAI MÔ HÌNH

3.1 Đề xuất mô hình

Để giải quyết bài toán phát hiện ảnh giả mạo, nhóm đề ra một mô hình gồm ba giai đoạn như hình ***Figure 7***.

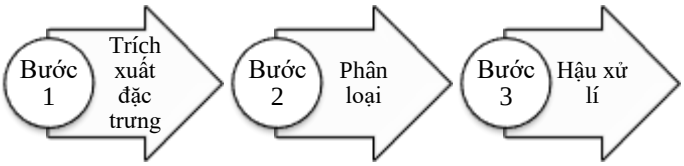


Figure 7 - Các giai đoạn xử lí

Giai đoạn đầu tiên là trích xuất đặc trưng, nhóm sử dụng cửa số trượt để chia ảnh thành những mảnh nhỏ có kích thước là 32x32. Tiếp theo, nhóm chuyển đổi ảnh từ kênh màu RGB sang kênh YCrCb bởi vì độ nhạy với những chi tiết bị làm giả ở kênh màu YCrCb là tốt hơn so với kênh màu thông dụng RGB [20]. Sau đó, áp dụng biến đổi Wavelets để trích xuất đặc trưng và vector hóa chúng thành một vector có 300 đặc trưng cho từng mảnh ảnh.

Giai đoạn thứ hai là phân loại, bởi vì các vector đặc trưng của ảnh gốc và ảnh giả mạo là không tuyến tính nên nhóm chọn mạng Nơ-ron nhân tạo gồm 6 lớp như hình ***Figure 8*** để phân loại giữa ảnh gốc và ảnh giả. Lớp ngõ vào của mạng Nơ-ron là những vector đặc trưng ở giai đoạn đầu nên lớp thứ nhất của mạng có chiều giống với chiều của vector đặc trưng, trong mỗi lớp sử dụng hàm số kích hoạt là hàm **Leaky ReLU** [25][26][27] và hệ số được khởi tạo ban đầu dùng phương pháp **Xavier** [24]. Cuối cùng của mạng Nơ-ron nhân tạo là một lớp **Softmax** để phân loại. Để tránh hiện tượng Overfitting, nhóm sử dụng Dropout [23] ở các lớp giữa.

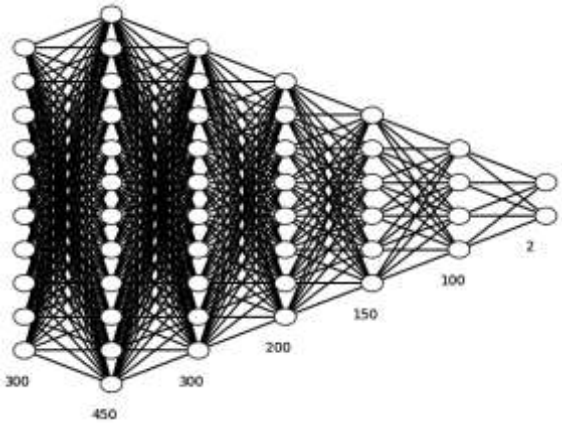


Figure 8 - Mạng Nơ-ron nhân tạo đề xuất

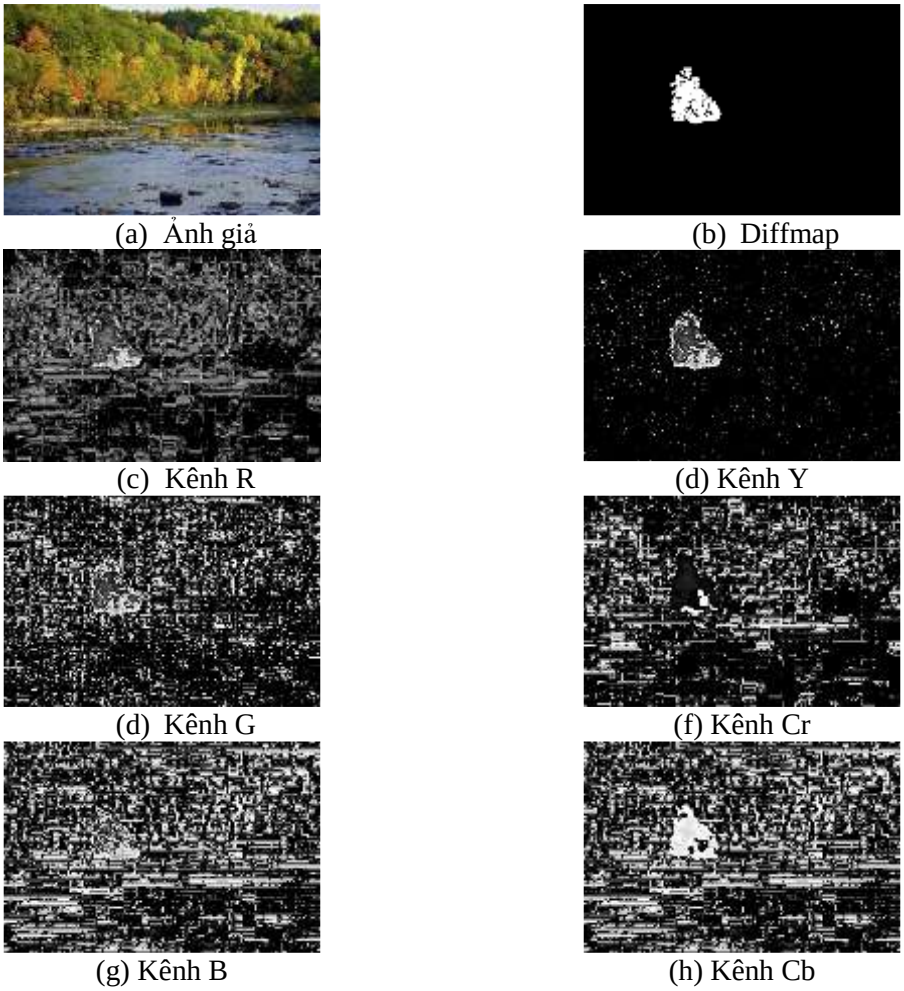
Giai đoạn thứ ba là hậu xử lí. Trong giai đoạn này, nhóm kiểm tra lại mảnh bằng cách xem xét các các mảnh xung quanh. Tính toán chỉ số tin cậy, nế chỉ số này vượt khỏi mức ngưỡng thì thì mảnh đó xem như là của ảnh giả và ngược lại là mảnh của ảnh thật.

3.2 Tập dữ liệu

Một trong những yếu tố quan trọng ảnh hưởng mạnh đến kết quả của quá trình phân loại sử dụng mạng Nơ-ron nhân tạo là tập dữ liệu. Tập dữ liệu chuẩn phải được thế giới công nhận, từ đó mới có thể đưa ra các đánh giá so với các phương pháp khác. Vì vậy, nhóm chọn tập dữ liệu CASIA-v2 [21], đây là tập dữ liệu về ảnh giả mạo được sử dụng rộng rãi trong các nghiên cứu khác trên thế giới. Ở tập CASIA v2 có 7491 ảnh là thật và 5123 ảnh bị làm giả, các ảnh này có nhiều kích cỡ từ 240x160 dến 900x600 với các định dạng là JPEG, BMP và TIFF. Trong số các ảnh giả, mỗi ảnh được tạo nên bằng cách lấy một phần của ảnh thật thứ nhất (con người, động vật, bầu trời) và ghép vào nền của ảnh thật thứ hai. Tên của mỗi ảnh giả có ghi rõ tên của hai ảnh thật tạo nên nó.

Bước đầu tiên, nhóm chia tập dữ liệu thành hai tập (một tập ảnh thật và một tập ảnh giả). Đối với mỗi ảnh giả, nhóm tìm ra vùng sai khác (***diffmap***) hình ***Figure 9(b)*** bằng cách lấy ảnh này trừ cho ảnh thật được sử dụng để làm nền, phép trừ được thực hiện ở kênh màu YCrCb và thực hiện bộ lọc hình thái học (morphological filter) vào lớp Y của ảnh. Trong hình ***Figure 9***, ***(a)*** là ảnh giả, các ảnh ***(c), (e), (g)*** là các ảnh sau khi thực hiện phép trừ trên kênh màu RGB. Tương tự, phép trừ thực hiện trên kênh màu YCrCb được thể hiện trên các ảnh (***d), (f), (h).***

Theo quan điểm của nhóm, sau khi dùng cửa sổ trượt để chia ảnh thành mảnh nhỏ, nếu những mảnh này nằm trọn trong vùng giả mạo thì không hỗ trợ được cho mạng Nơ-ron nhân tạo phía sau để nhận ra được những sự thay đổi trong ảnh giả. Do đó, thay vì chọn những mảnh bên trong vùng giả mạo, nhóm chọn những mảnh chỉ chứa rìa của vùng giả, tức là chứa đường biên giữa vùng thật và giả như hình ***Figure 10***. Nhờ vậy, mạng Nơ-ron nhân tạo có thể phát hiện được sự đối lập giữa hai vùng. Ngoài ra, tập dữ liệu phải là tập cân bằng, tức là có số mảnh giả bằng với số mảnh thật, nhưng số mảnh giả khá ít nên nhóm sử dụng biến đổi hình học để tăng số lượng tập mảnh giả. Đối với mỗi mảnh giả, có thể tạo ra được thêm ba mảnh mới bằng cách xoay 90°, 180°, và 270°. Để xây dựng tập dữ liệu, nhóm chỉ sử dụng 1500 trong số 5123 ảnh giả. Ngược lại, tạo tập dữ liệu của những mảnh thật dễ dàng hơn, chỉ cần chọn một cách ngẫu nhiên một mảnh trong bức ảnh thật.



(a) Ảnh giả (b) Diffmap

(c) Kênh R (d) Kênh Y

(d) Kênh G (f) Kênh Cr

(g) Kênh B (h) Kênh Cb

Figure 9 - So sánh hiệu quả trên hai kênh màu

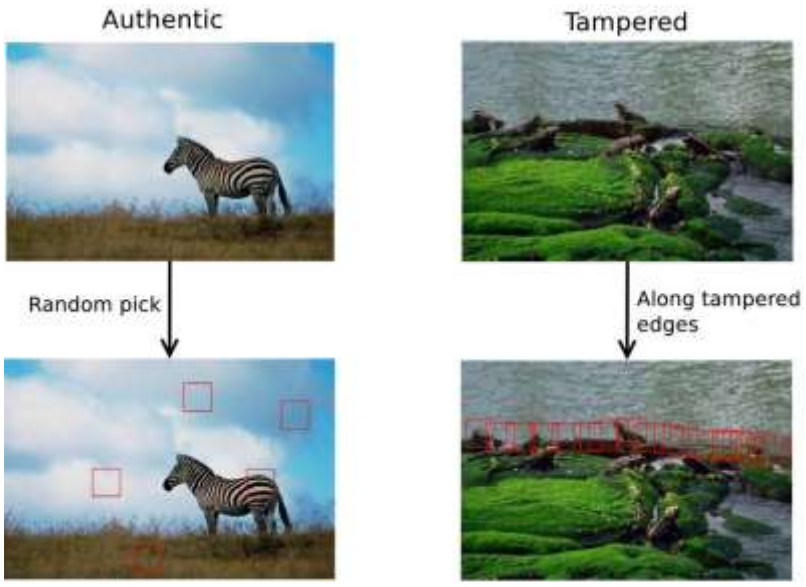


Figure 10 - Tạo tập dữ liệu những mảnh của ảnh thật và ảnh giả

3.3 Trích xuất đặc trưng

Với mỗi mảnh ảnh trong tập dữ liệu, nhóm chuyển chúng về kênh màu YCrCb, sau đó sử dụng biến đổi Daubechies Wavelets (db1-db5) cho mỗi lớp của mảnh YCrCb. Công việc này tạo ra 150 ma trận (3 kênh x 5 biến đổi x 10 ma trận kết quả). Mỗi ma trận, nhóm tính trung bình, phương sai và tổng. Kết quả là nhóm thu được vector đặc trưng có chiều dài 450 cho mỗi mảnh.

Để có cái nhìn tổng quát về điểm khác biệt của dữ liệu, nhóm phân tích vector đặc trưng. Đầu tiên, chuẩn hóa dữ liệu:

$$z = \frac{x - \mu}{\Delta} \tag{4.1}$$

Trong đó $\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$ và $\Delta = x_{max} - x_{min}$. Tiếp theo, nhóm tính vector trung bình và phương sai của hai tập dữ liệu của hai lớp $(\mu_1, \mu_2, \sigma_1, \sigma_2)$. Cuối cùng, vector phân biệt được tính theo công thức:

$$dr = \frac{(\mu_1 - \mu_2)^2}{{\sigma_1}^2 + {\sigma_2}^2} \tag{4.2}$$

Bởi vì nhóm mong muốn tập dữ liệu ảnh thật phân biệt với dữ liệu trong tập ảnh giả nên các thành phần trong vector phân biệt càng lớn càng tốt.

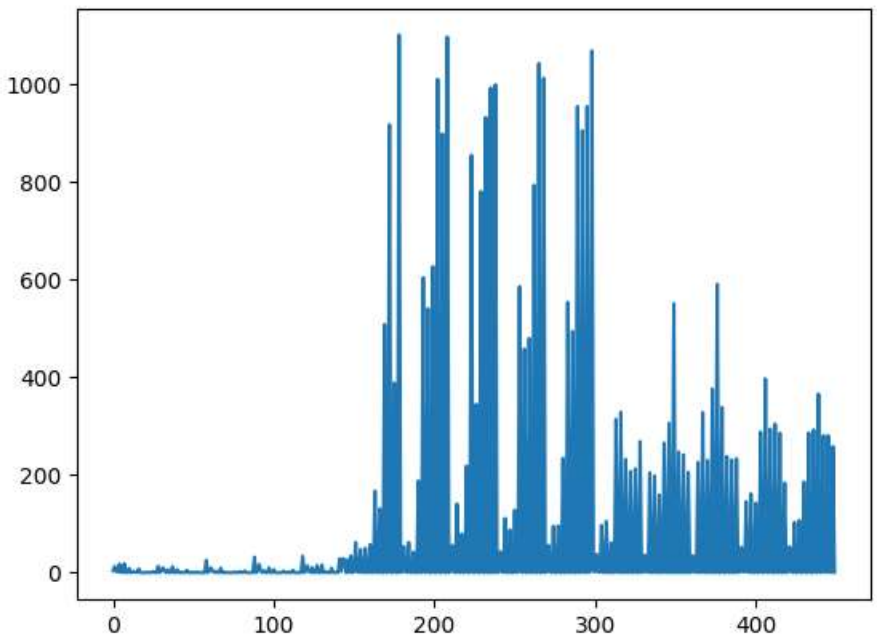


Figure 11 - Vector phân biệt

Hình ***Figure 11*** thể hiện kết quả của công thức 4.2. Ở đây có thể thấy, 150 thành phần đầu tiên không mang nhiều ý nghĩa, trong khi đó phần còn lại thì cao hơn hẳn. Khi xem xét 300 thành phần còn lại của vector dr, thì dữ liệu là khá phân biệt. Mặt khác, các thành phần không mang nhiều ý nghĩa thuộc kênh Y. Do đó, phương pháp trích xuất đặc trưng, biến đổi Daucechies Wavelets cho kênh Y sẽ không có đóng góp cho mạng việc dào tạo mạng Nơ-ron nhân tạo, vì vậy nhóm loại bỏ 150 thành phần dâu tiên này.

3.4 Phân loại

Nhóm sử dụng mạng Nơ-ron nhân tạo được giới thiệu ở hình ***Figure 8*** để phân loại. Đầu tiên, nhóm khởi tạo các trọng số và các hệ số tự do với phương pháp Xavier để tăng tốc độ hội tụ của mạng. Thêm nữa, ở các lớp giữa, nhóm sử dụng hàm số kích hoạt là hàm **Leaky ReLU** để cho mạng Nơ-ron phân loại tốt hơn các dữ liệu không tuyến tính. Dữ liệu các mảnh của ảnh mà nhóm thu thập được là 399046 mảnh, trong đó có 198520 là của ảnh giả và 200526 là của ảnh thật. Dữ liệu ở đây là khá cân bằng nên nên mạng Nơ-ron sẽ không có khuynh hướng nghiêng về một phần. Sau đó, tập các mảnh này được chia thành hai phần (một cho huấn luyện và một cho đánh giá). 90% của tập dữ liệu sẽ được đưa vào tập huấn luyện, rang buộc số mảnh của ảnh thật và giả phải bằng nhau. Sau đó, phần còn lại sẽ đưa vào tập đánh giá. Ở đây, nhóm thay tập kiểm tra (test set) bằng tập đánh giá (evaluating set) bởi vì nhóm chỉ tập trung vào việc huấn luyện mạng Nơ-ron nhân tạo để tìm ra các thông số tối ưu nhất. Do đó, tập đánh giá sẽ giúp nhóm đánh giá các hệ số một cách khác quan. Ngoài ra, tập kiểm tra sẽ được tạo và sử dụng trong quá trình kiểm tra, sau khi quá trình huấn luyện đã xong.

Trước khi huấn luyện, nhóm thực hiện chuẩn hóa dữ liệu để tất cả các vector đặc trưng đều nằm trong một khoảng đều nhau, điều này sẽ giúp cho mạng nơ-ron học các đặc trưng dễ dàng hơn và đồng đều hơn. Bên dưới là vector trung bình và phương sai của tập huấn luyện:

$$x_{mean} = \frac{1}{N_{train}} \sum_{i=1}^{N} x_i^{(train)} \tag{4.3}$$

$$x_{var} = \sqrt{\frac{1}{N_{train}} \sum_{i=1}^{N} (x_i - x_{mean})^2} \tag{4.4}$$

Từ hai công thức trên, tập huấn luyện được chuẩn hóa:

$$X_{train} = \frac{x_{train} - x_{mean}}{x_{var}} \quad (4.5)$$

Từ đây, trung bình và phương sai được tính trong công thức 4.3 và 4.4 được lưu trên đĩa cứng như là các ***siêu thông số (hyper-parameter)*** của mạng Nơ-ron nhân tạo. Khi thực hiện kiểm tra hoặc đánh giá, nhóm dùng hai vector đó để chuẩn hóa dữ liệu kiểm tra và đánh giá. Cuối cùng, thực hiên huấn luyện mạng Nơ-ron bằng ngôn ngữ lập trình Python và thư viện Tensorflow trên máy tính chip Quad-Core-i7, 8GB DDR4 RAM và GPU NVIDIA GEFORCE GTX 1050.

3.5 Hậu xử lí

Bước cuối cùng này chỉ sử dụng để kiểm tra. Đầu tiên, nhóm chọn thủ công 757 ảnh thật và 800 ảnh giả, những ảnh này riêng biệt so với tập huấn luyện. Đối với mỗi ảnh, sử dụng một ***cửa sổ trượt (sliding window)*** với ***bước nhảy (stride)*** là 16 để lấy ra những mảnh 32x32. Sau đó, những mảnh được chuyển sang kênh màu YCrCb và vector đặc trưng được trích xuất bằng biến đổi Daucechies Wavelets. Sau khi qua 6 lớp của mạng Nơ-ron, một danh sách các nhãn tương ứng với các mảnh xuất hiện ở lớp ra (output). Tiếp theo, quá trình hậu xử lí được áp dụng để lọc ra những những mảnh ảnh giả, những mảnh mà không nhận ra, dựa trên thông tin của các nhãn xung quanh.

Với mỗi mảnh, có nhiều nhất 8 mảnh lân cận (các mảnh tại viền và góc của ảnh thì có thể ít hơn). Giả sử mảnh p_0 có nhãn là $l(p_0) = 1$, có k mảnh lân cận là $p_i(i = 1, \dots, k)$, nên chỉ số tin cậy có thể được tính theo công thức:

$$Reliability = \frac{1}{k+1}\sum_{i=0}^{k} l(p_i) \quad (4.6)$$

Từ đây, nếu chỉ số tin cậy vượt qua một mức ngưỡng α $(0 < \alpha < 1)$ thì nhãn của mảnh đó được chấp nhận, nếu chỉ số tin cậy thấp hơn thì nhãn của mảnh đó chuyển thành ảnh thật. Nếu tất cả các mảnh trong một ảnh là thật thì ảnh đó là ảnh chưa qua chỉnh sửa, ngược lại, nếu có ít nhất một mảnh của ảnh là giả thì ảnh đó có qua chỉnh sửa.

Chương 4: KẾT QUẢ THỰC NGHIỆM

Trong phần này, nhóm định nghĩa những hệ số để đánh giá cho mô hình. Kết quả của quá trình phân loại có bốn trường hợp:

- True Positive (TP): là những mảnh của ảnh giả được phân loại đúng.
- True Negative (TN): là những mảnh ảnh thật được phân loại đúng.
- False Positive (FP): là những mảnh của ảnh giả nhưng bị phân loại là ảnh thật.
- False Negative (FP): là những mảnh của ảnh thật nhưng bị phân loại là ảnh giả.

Những công thức để đánh giá mô hình:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.1}$$

$$Precision = \frac{TP}{TP + FP} \tag{5.2}$$

$$Recall = \frac{TP}{TP + FN} \tag{5.3}$$

$$Fscore = \frac{2 * Precision * Recall}{Precision + Recall} \tag{5.4}$$

Trong bốn công thức trên, Accuracy thể hiện cho hiệu năng chung cho toàn mô hình. Tuy nhiên, nếu mô hình có tất cả đầu vào là ảnh thật thì độ chính xác ở ngõ ra sẽ là rất cao. Do đó, hai hệ số Precision và Recall được đưa ra để bổ sung cho hạn chế của hệ số Accuracy. Để rõ ràng hơn, Precision phản ánh số lượng mẫu làm mảnh giả được phân loại đúng trong số các mảnh giả, hệ số Precision cao đồng nghĩa với độ chính xác của mảnh tìm được là cao. Recall là tỉ lệ số mảnh giả được phân loại đúng trong số các mảnh được phân loại là giả, tức là tỉ lệ bỏ sót các mảnh thực sự là mảnh giả. Mặt khác, nhóm muốn có một số liệu duy nhất đánh giá cho khả năng sai lệch của mô hình. Vậy nên hệ số Fscore bao gồm cả thông tin của Precision và Recall.

4.1 Quá trình huấn luyện

Trong quá trình huấn luyện, mạng Nơ-ron nhân tạo tự học và tối ưu các hệ số để phân loại. Ở đây nhóm sử dụng Adam optimizer [28] với tốc độ học (learning rate) là $8e^{-4}$. Để phù hợp với phần cứng, nhóm chọn kích thước của mỗi tập mảnh ảnh là 512 mảnh. Sau 14100 vòng lặp, nhóm thu được kết quả như hình ***Figure 12***.

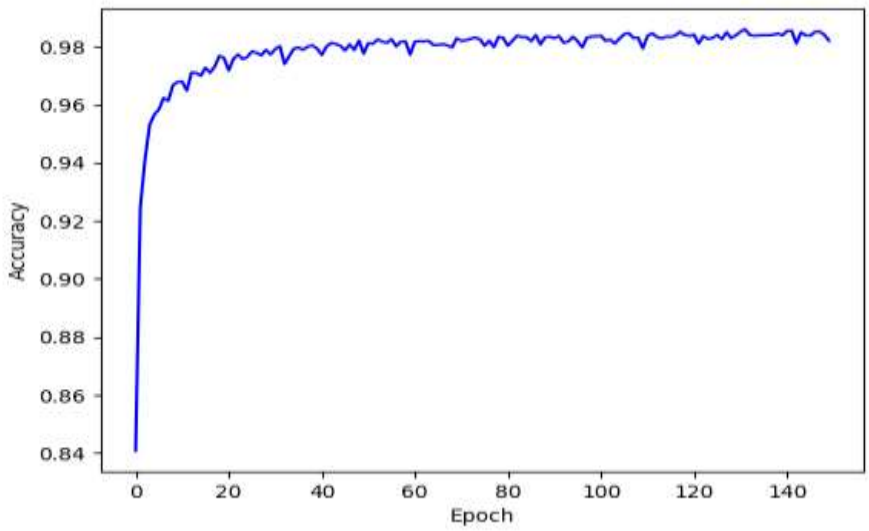


Figure 12- Độ chính xác suốt quá trình huấn luyện

Hệ số	Giá trị
Accuracy	98.21%
Precision	99.08%
Recall	97.32%
Fscore	98.19%

Table 1 - Hệ số ở bước cuối cùng trong quá trình huấn luyện

Bảng ***Table 1*** cho thấy kết quả huấn luyện mạng Nơ-ron nhân tạo. Có thể thấy được các hệ số rất cao, điều đó thể hiện sức mạnh của mạng Nơ-ron này. Thời gian để thực hiện quá trình huấn luyện thường ít hơn 15 phút.

4.2 Quá trình kiểm tra

Sau khi quá trình huấn luyện hoàn tất, nhóm xây dựng mô hình kiểm tra. Nhóm sử dụng 757 ảnh giả và 800 ảnh thật cho bước này. Đầu tiên, sử dụng cửa sổ trượt để lấy ra những mảnh trong bức ảnh, tiếp theo chuyển ảnh sang kênh màu YCrCb và sử dụng biến đổi Wavelets để trích xuất vector đặc trưng. Sau đó, đưa các vector vào mạng Nơ-ron nhân tạo để dự đoán mảnh ấy là thật hay giả, một bước hậu xử lí được áp dụng để dự doán cho bức ảnh là thật hay giả. Cuối cùng, nhóm thu được kết quả ở bảng ***Table 2***. Những số liệu này được tính toán trên đơn vị là ảnh, khác với đơn vị là mảnh ảnh trong quá trình huấn luyện. Kết quả ở bảng ***Table 2*** có sự khác nhau so với bảng ***Table 1*** là do quá trình kiểm tra có một bước hậu xử lí-tổng hợp nhãn, nên kết quả ở bảng ***Table 2*** được đánh giá với đơn vị là ảnh, không phải là các mảnh ảnh.

Hệ số	Giá trị
Accuracy	97.11%
Precision	98.88%
Recall	95.65%
Fscore	97.23%

Table 2 - Hệ số trong quá trình kiểm tra

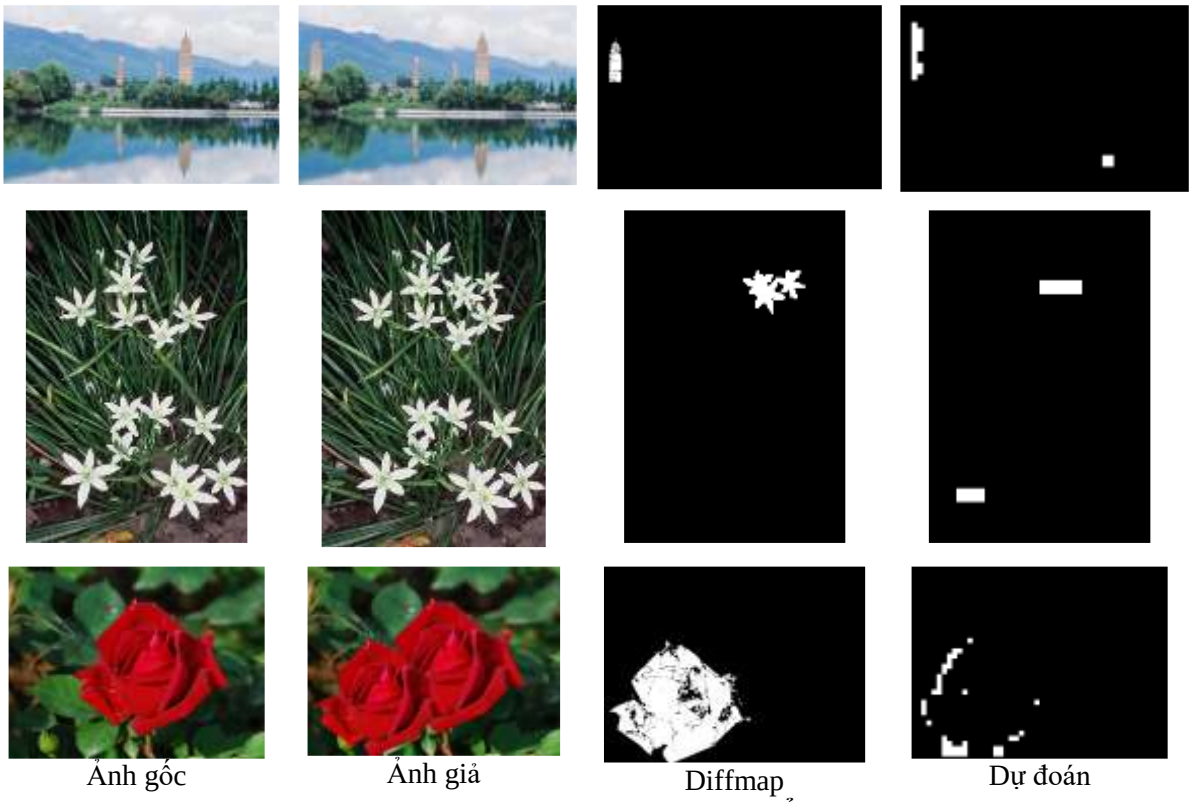


Figure 13 - Dự đoán của quá trình kiểm tra

Hình ***Figure 13*** cho thấy kết quả của quá trình kiểm tra. Có thể thấy kết quả dự đoán khá phù hợp với diffmap. Do các mảnh của ảnh giả được đưa vào mạng Nơ-ron nhân tạo mang thông tin của phần rìa của vùng thật và giả nên dự đoán sẽ đánh dấu vào các vùng rìa. Ví dụ như hàng thứ ba của ảnh trên, phần dự đoán của quá trình kiểm tra chỉ đánh dấu vào phần rìa của hoa hồng được dán thêm vào ảnh.

4.3 Quá trình đánh giá

Ngoài mạng Nơ-ron nhân tạo với ngõ vào là vector đặc trưng có chiều dài 300 trong hình ***Figure 8***, nhóm còn thử nghiệm mạng Nơ-ron khác như trong hình Mạng Nơ-ron này giống với mạng Nơ-ron trước, ngoại trừ ngõ vào là vector có 450 đặc trưng. Nhóm gọi mạng Nơ-ron trong hình ***Figure 8*** là input-300 và mạng Nơ-ron trong hình ***Figure 14*** là input-450. Mục đích của việc thử nghiệm này là chứng minh mô hình input-300 giảm được kích thước

nhưng hiệu năng hoạt động vẫn tốt. Hình ***Figure 15*** vẽ hai đồ thị của độ chính xác trong suốt quá trình huấn luyện của hai mô hình. Thêm vào đó, ***Table 3*** ghi lại một số giá trị của quá trình huấn luyện và kiểm tra. Quá trình kiểm tra của mô hình input-300 tốt hơn so với input-450. Sự vượt trội này là do mô hình input-300 học được các đặc trưng một cách tổng quan và khái quát, tránh việc ***học kĩ (over-fitting)***. Do đó, bằng cách giảm đặc trưng, mô hình input-300 vẫn duy trì độ chính xác và tốc độ tính toán được cải thiện.

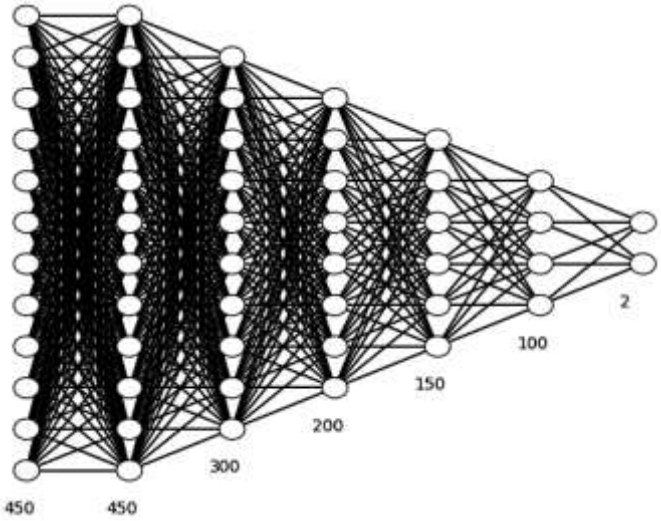


Figure 14 - Mô hình input-450

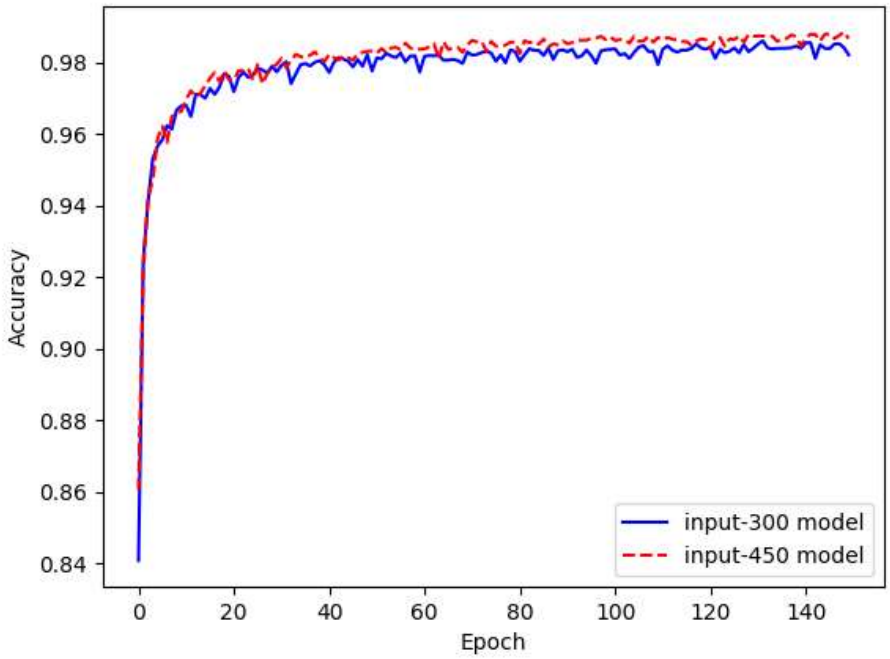


Figure 15 - Độ chính xác của input-300 và input-450

Hệ số	Quá trình huấn luyện		Quá trình kiểm tra	
	Input-300	**Input-450**	**Input-300**	**Input-450**
Accuracy	98.21%	98.68%	**97.11%**	**96.34%**
Precision	99.08%	99.00%	98.88%	97.62%
Recall	97.32%	98.31%	95.65%	95.36%
Fscore	98.19%	98.65%	97.23%	96.48%
Thời gian	**14m25s**	**15m55s**	**3.570s**	**3.757s**

Table 3 - So sánh giữa hai mô hình input-300 và input-450

Bên cạnh đó, sau khi quá trình huấn luyện của input-450 hoàn tất, nhóm tiến hành xem xét các trọng số lớp thứ nhất trong mạng nơ-ron, tương ứng với mội trọng số là mỗi đặc trưng. Hình ***Figure 16*** thể hiện độ lớn của các trọng số của lớp thứ nhất này. Trong đó, trục features tương ứng với 450 phần tử của vector đặc trưng đầu vào, còn trục omega tương ứng với 450 nơ-ron nằm ở lớp theo sau đó. Ta có thể thấy rằng các trọng số của 150 đặc trưng đầu tiên là rất thấp, gần như bằng 0, điều này có nghĩa các đặc trưng này không có tác dụng trong việc phân loại. Trong khi đó, ở 300 đặc trưng tiếp theo, giá trị của các trọng số lớn hơn hẳn, chứng tỏ các trọng số này có "ảnh hưởng" hơn trong việc nhận dạng. Điều này càng làm rõ hơn tính thích hợp khi loại bỏ 150 đặc trưng đầu tiên.

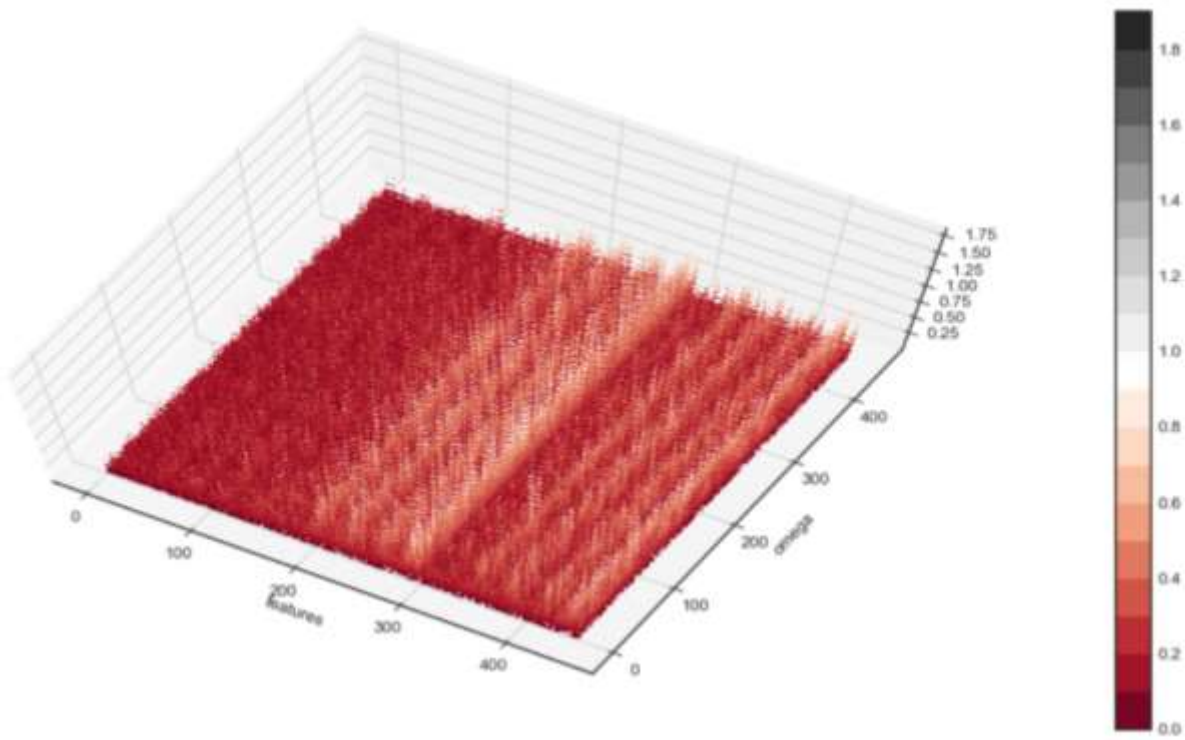


Figure 16 - Độ lớn của các trọng số lớp đầu tiên sau khi huấn luyện của mô hình input-450

4.4 So sánh với các mô hình khác

Sau khi kiểm tra mô hình, nhóm tiến hành so sánh hiệu năng so với các phương pháp khác. Để làm rõ tính hiệu quả của mô hình hướng dữ liệu so với các mô hình truyền thống, nhóm chọn hai mô hình thông thường [29] [30] và thêm một mô hình theo hướng dữ liệu [17]. Thêm vào đó, các phương pháp này đánh giá độ chính xác theo đơn vị ảnh, điều này sẽ làm cho việc đánh giá trở nên công bằng, vì phương pháp của nhóm cũng đánh giá dựa trên ảnh. Việc so sánh dựa trên độ chính xác khi kiểm tra trên tập dữ liệu CASIA v2. Trong bảng ***Table 4***, phương pháp nhóm đề xuất xếp vị trí thứ hai, vượt qua hai phương pháp thông thường. Nói một cách chính xác, mô hình [17], mô hình của nhóm và mô hình thông thường [29] có độ chính xác xấp xỉ, trong khi đó mô hình thông thường [30] thì thấp hơn nhiều so với nhóm đầu. So sánh này cho thấy mô hình hướng dữ liệu vượt trội hơn môn hình thông thường.

Tiếp theo, nhóm cũng so sánh và nhận xét hai mô hình chuyển hướng dữ liệu (của [17] và của nhóm đề xuất). Trong mô hình [17], Rao *et al.* sử dụng một mạng Nơ-ron xoắn (Convolutional Neural Network), sử dụng phân loại SVM ở lớp cuối cùng, và họ sử dụng toàn bộ các ảnh trong tập dữ liệu CASIA v2 để huấn luyện và kiểm tra. Trong khi đó, do tạo dữ liệu bằng phương pháp thủ công nên tập dữ liệu của nhóm nhỏ hơn rất nhiều so với tập dữ liệu của mô hình [17]. Ngoài ra, mô hình của nhóm cũng cạn hơn so với mạng Nơ-ron xoắn 10 lớp của mô hình [17]. Tuy nhiên, độ chính xác của hai mô hình gần như bằng nhau. Điều này chứng tỏ mô hình của nhóm có thể chịu được tập dữ liệu nhỏ. Mặt khác, do mô hình cạn của nhóm nên thời gian thực hiện nhanh hơn trong quá trình huấn luyện cũng như là kiểm tra.

Mô hình	Độ chính xác	Số ảnh sử dụng	Số nơ-ron
Rao et al. [17]	97.83%	2102	606752
Nhóm đề xuất	**97.11%**	**1557**	**1502**
Goh et al. [29]	96.21%	1200	-
He et al. [30]	87.37%	1200	-

Table 4 - So sánh với các mô hình khác

Chương 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Kết luận

Qua báo cáo này, nhóm đề xuất một mô hình có sai số thấp và theo hướng dữ liệu để giải quyết vấn đề phát hiện ảnh giả mạo. Mạng Nơ-ron nhân tạo cùng với các thành phần (hàm số kích hoạt, khởi tạo trọng số và hệ số tự do, phương pháp tối ưu, chuẩn hóa) đươc thiết kế để phát hiện ảnh giả. Bằng việc xem xét các đặc trưng được trích xuất, nhóm đã chỉ ra được rằng không phải toàn bộ đặc trưng là cần thiết, nên có thể giảm sai số tính toán khi chỉ sử dụng các đặc trưng quan trọng. Qua kết quả thử nghiệm, mô hình của nhóm có thể gặp khó khăn trong một số điều kiện, như là tập dữ liệu ảnh giả ít và giới hạn về phần cứng. Mô hình của nhóm phát hiện được ảnh gải với độ chính xác cao là 97.11%, trong đó, mô hình có thể đạt được các hệ số Precision và Recall lần lượt là 98.88% và 95.65%.

5.2 Hướng phát triển

Trong thời gian tới, nhóm sẽ tiếp tục nghiên cứu các loại đặc trưng khác mà có sự phân biệt rõ ràng hơn giữa ảnh thật và ảnh giả. Điều này sẽ giúp cho việc nhận dạng trở nên chính xác hơn. Nhóm cũng sẽ xem xét việc giảm các đặc trưng không cần thiết để tăng tốc độ tính toán cho mô hình. Bên cạnh đó, việc sử dụng cửa sổ dịch tốn khá nhiều thời gian để tính toán. Nên nhóm quyết định đầu tư tìm hiểu về một số phương pháp Đề xuất khu vực (Region Proposal) để có thể rút ngắn thời gian tính toán hơn nữa. Ngoài việc sử dụng mạng Nơ-ron nhân tạo, nhóm sẽ quan tâm, ngiên cứu thêm các loại hình Deep Learning khác để tăng hiệu năng của quá trình phân loại, như là mạng Nơ-ron xoắn (CNN), và Bộ nhớ ngắn hạn dài (Long Short Term Memory).

TÀI LIỆU THAM KHẢO

[1] X. Pan and S. Lyu, *"Region duplication detection using image feature matching"*, IEEE Transactions on Information Forensics and Security, vol. 5, no.4, ISSN: 1556-6013, pp. 857-867, 2010.

[2] I. Amerini, L. Ballan, R. Caldelli, A. Del Bimbo and G.Serra, *"A sift-based forensic method for copymove attack detection and transformation recovery"*, IEEE Transactions on Information Forensics and Security, vol.6, no. 3, ISSN**:** 1556-6013, pp. 1099-1110, 2011.

[3] P. Kakar, N. Sudha, *"Exposing postprocessed copy-paste forgeries through transform-invariant feature"*, IEEE Transactions on Information Forensics and Security, vol. 7, no. 3, ISSN**:** 1556-6013, pp. 1018-1028, June 2012.

[4] T. Le-Tien, T. Huynh-Kha, L. P.-Cong-Hoan, A. Tran-Hong, N. Dey, M. Luong, "*Combined Zernike Moment and Multiscale Analysis for Tamper Detection in Digital Images*", Informatica (An International Journal of Computing and Informatics), Vol.41, No.1, March 2017, ISSN: 0350-5596.

[5] S.-J. Ryu, M.-J. Lee and H.-K. Lee, *"Detection of copy-rotate-move forgery using Zernike moments"*, Information Hiding Conference, Lecture Notes in Computer Science, vol. 6387, Springer, Heidelberg-Berlin, 2010, ISBN: 978-3-642-16434-7.

[6] H.-J. Lin, C.-W. Wang and Y.-T. Kao, *"Fast copy-move forgery detection"*, WSEAS Transactions on Signal Processing, vol. 5, no. 5, ISSN: 0031-3203, pp. 188-1975, 2009.

[7] V. Christlein, C. Riess, J. Jordan and E. Angelopoulou, *"An evaluation of popular copy-move forgery detection approaches"*, IEEE Transactions on Information Forensics and Security, vol. 7, no. 6, ISSN**:** 1556-6013, pp. 1841-1854, 2012.

[8] Z. Lin, J. He, X. Tang, K. Tang, *"Fast, automatic and fine-grained tampered jpeg image detection via DCT coefficient analysis"*, Pattern Recognition, vol. 42, no. 11, ISSN: 0031-3203, pp. 2492-2501, January 2009.

[9] W. Wang, J. Dong, T. Tan, *"Exploring DCT coefficient quantization effects for local tampering detection"*, IEEE Transactions on Information Forensics and Security, vol. 9, no. 10, ISSN**:** 1556-6013, pp. 1653-1666, October 2014.

[10] L. Chen, T. Hsu, *"Detecting recompression of JPEG images via periodicity analysis of compression artifacts for tampering detection"*, IEEE Transactions on Information Forensics and Security, vol. 6, no. 2, ISSN**:** 1556-6013, pp. 396-406, June 2011.

[11] L. Thing, Y. Chen, C. Cheh, *"An improved double compression detection method for JPEG image forensics"*, In IEEE International Symposium on Multimedia, pages 290-297, December 2012, ISBN: 978-1-4673-4370-1.

[12] F. Zach, C. Riess, and E. Angelopoulou, *"Automated image forgery detection through classification of JPEG ghosts"*, Pattern Recognition. DAGM/OAGM

2012. Lecture Notes in Computer Science, vol 7476, Springer, Berlin, Heidelberg, 2012, ISBN: 978-3-642-32716-2.

[13] T. Bianchi, A. Piva, *"Image forgery localization via block-grained analysis of jpeg artifacts"*, IEEE Transactions on Information Forensics and Security, vol. 7, no. 3, ISSN**:** 1556-6013, pp. 1003-1017, June 2012.

[14] T. Bianchi, A. Piva, *"Image forgery localization via block-grained analysis of JPEG artifacts"*, IEEE Transactions on Information Forensics and Security, vol. 7, no. 3, ISSN**:** 1556-6013, pp. 1003-1017, June 2012.

[15] J. Chen, X. Kang, Y. Liu and Z. J. Wang, *"Median Filtering Forensics Based on Convolutional Neural Networks"*, IEEE Signal Processing Letters, vol. 22, no. 11, pp. 1849-1853, ISSN: 1070-9908, November 2015.

[16] B. Bayar, M. C. Stamm, *"A Deep Learning Approach to Universal Image Manipulation Detection Using a New Convolutional Layer"*, Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security, pp. 5-10, New York-USA, 2016, ISBN: 978-1-4503-4290-2.

[17] Rao Yuan, Ni Jiangqun, *"A deep learning approach to detection of splicing and copy-move forgeries in images"*, IEEE International Workshop on Information Forensics and Security (WIFS), Abu Dhabi-United Arab Emirates, 2016, ISBN: 978-1-5090-1139-1.

[18] J.Ouyang, Y.Liu, M.Liao, *"Copy-Move Forgery Detection Based on Deep Learning"*, 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics, Shanghai-China, 2017, ISBN: 978-1-5386-1938-4.

[19] Y. Zhang, J. Goh, L. Win, V. Thing, *"Image Region Forgery Detection: A Deep Learning Approach"*, Proceedings of the Singapore Cyber-Security Conference, Singapore, 2016, ISBN: 978-1-61499-616-3.

[20] W. Wang, J. Dong, and T. Tan, *"Effective image splicing detection based on image chroma"*, IEEE International Conference on Image Processing, pp. 1257-1260, Cairo-Egypt, 2009, ISBN: 978-1-4244-5653-6.

[21] J. Dong and W. Wang, "Casia tampering detection dataset", 2011.

[22] A. Krizhevsky, I. Sutskever, G. Hinton, *"Imagenet classification with deep convolutional neural networks"*, NIPS'12 Proceedings of the 25th International Conference on Neural Information Processing Systems, vol. 1, pp. 1097-1105, Nevada-USA, 2012, DOI: 10.1145/3065386.

[23] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, *"Dropout: a simple way to prevent neural networks from overfitting"*, The Journal of Machine Learning Research, vol. 15, no. 1, ISSN 1533-7928, pp. 1929-1958, January 2014.

[24] Xavier Glorot, Yoshua Bengio, *"Understanding the difficulty of training deep feedforward neural networks"*, Proceedings of the 13rd International

Conference on Artificial Intelligence and Statistics, PMLR 9, pp. 249-256, Sardinia-Italy, 2010, http://proceedings.mlr.press/v9/glorot10a/glorot10a.pdf.

[25] V. Nair, E. Hinton, *"Rectified Linear Units Improve Restricted Boltzmann Machines"*, Proceedings of the 27th International Conference on Machine Learning, pp. 807-814, Haifa-Israel, 2010, ISBN: 978-1-60558-907-7.

[26] B. Xu, N. Wang, T. Chen, M. Li, *"Empirical Evaluation of Rectified Activations in Convolutional Network"*, https://arxiv.org/abs/1505.00853v2, 2015.

[27] K. He, X. Zhang, S. Ren, J. Sun *"Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification"*, https://arxiv.org/abs/1502.01852v1, 2015.

[28] P. Kingma, J. Ba, *"Adam: A Method for Stochastic Optimization"*, 3rd International Conference for Learning Representations, San Diego-USA, 2015, https://arxiv.org/abs/1412.6980.

[29] J. Goh and V. L. L. Thing, *"A hybrid evolutionary algorithm for feature and ensemble selection in image tampering detection"*, International Journal of Electronic Security and Digital Forensics, vol. 7, no. 1, ISSN: 1751-911X, pp.76-104, March 2015.

[30] Z. He, W. Lu, W. Sun, J. Huang, *"Digital image splicing detection based on Markov features in DCT and DWT domain"*, Pattern Recognition, vol. 45, no. 12, ISSN: 0031-3203, pp. 4292-4299, 2012.

PHỤ LỤC

Chương trình trích xuất đặc trưng

```
#-------------------------------------------------------------------------------
#   Import library
#-------------------------------------------------------------------------------
from utils import patches, basic_ft
from multiprocessing import Pool, cpu_count


#-------------------------------------------------------------------------------
#   Define
#-------------------------------------------------------------------------------
# Files contain information of patches
POS_PATCHES_PATH = "label-data/patches-pos-2.txt"
NEG_PATCHES_PATH = "label-data/patches-neg-2.txt"

# Files contain extracted features
POS_FT_PATH = "dataset/pos-features.mat"
NEG_FT_PATH = "dataset/neg-features.mat"


#-------------------------------------------------------------------------------
#   Main execution
#-------------------------------------------------------------------------------
# Create parallel pool
pool = Pool(processes=cpu_count())


# Extract positive features
print("Collecting positive patches...")
patches_pos = patches.get_patches(POS_PATCHES_PATH, geo_trans=True)
print("Number of positive patches:", len(patches_pos))

print("Extracting positive features...")
features_pos = basic_ft.extract_all(patches_pos, pool=pool)
print("Number of positive features:", len(features_pos))
basic_ft.save(features_pos, POS_FT_PATH)
print("Positive features are saved in", POS_FT_PATH)
del patches_pos, features_pos


# Extract negative features
print("Collecting negative patches...")
patches_neg = patches.get_patches(NEG_PATCHES_PATH, geo_trans=False)
print("Number of negative patches:", len(patches_neg))

print("Extracting negative features...")
features_neg = basic_ft.extract_all(patches_neg, pool=pool)
print("Number of negative features:", len(features_neg))
basic_ft.save(features_neg, NEG_FT_PATH)
print("Negative features are saved in", NEG_FT_PATH)


# Close parallel pool
pool.close()
pool.terminate()
```

Chương trình huấn luyện mạng Nơ-ron học

```
#--------------------------------------------------------------------------------
#   Import library
#--------------------------------------------------------------------------------
import os
from scipy.io import savemat
import tensorflow as tf
tf.logging.set_verbosity(tf.logging.INFO)
from utils import model, basic_ft, data


#--------------------------------------------------------------------------------
#   Define
#--------------------------------------------------------------------------------
# Files contain extracted features
FT_POS_FPATH = 'dataset/pos-features.mat'
FT_NEG_FPATH = 'dataset/neg-features.mat'

# Files contain training and evaluating dataset
TRAIN_SET_FPATH = "dataset/train-set.mat"
EVAL_SET_FPATH = "dataset/eval-set.mat"
CONFIG_FPATH = "dataset/config-params.mat"

# Folder contains training information
LOG_PATH = 'log/nn/'

# Files contain training metrics
METRICS_300_FPATH = "train-metrics-300.mat"
METRICS_450_FPATH = "train-metrics-450.mat"

# Training configuration parameters
BATCH_SIZE = 512
EPOCH_NUM = 100
FT_NUM = 300


#--------------------------------------------------------------------------------
#   Main execution
#--------------------------------------------------------------------------------
def main(unused_argv):
    # Continue training the existing model
    list_paths = os.listdir(LOG_PATH)
    if len(list_paths) > 2:
        print("\nLoad data from disk\n")
        X_train, y_train = data.load(TRAIN_SET_FPATH)
        X_eval, y_eval = data.load(EVAL_SET_FPATH)

    # Start training a new model
    else:
        print("\nProcess data from disk\n")
        features_pos = basic_ft.load(FT_POS_FPATH)
        features_neg = basic_ft.load(FT_NEG_FPATH)

        features, N = basic_ft.compose(features_pos, features_neg)
        features, config_params = basic_ft.normalize(features)
        features_pos, features_neg = basic_ft.decompose(features, N)
        del features

        labels_pos = data.create_labels(features_pos, 1)
        labels_neg = data.create_labels(features_neg, 0)

        features_pos, labels_pos = data.shuffle(features_pos, labels_pos)
        features_neg, labels_neg = data.shuffle(features_neg, labels_neg)

        X_train, y_train, X_eval, y_eval = data.separate_train_test(
```

```
                                                features_pos, labels_pos,
                                                features_neg, labels_neg,
                                                ratio=0.9)
        data.save(X_train, y_train, TRAIN_SET_FPATH)
        data.save(X_eval, y_eval, EVAL_SET_FPATH)
        basic_ft.save(config_params, CONFIG_FPATH)
        del features_pos, labels_pos, features_neg, labels_neg, config_params

    # Modify data and choose model, based on number of features
    if FT_NUM==300:
        X_train = X_train[...,150:]; X_eval = X_eval[...,150:]
        METRICS_FPATH = METRICS_300_FPATH
        MODEL = model.neural300_net
    elif FT_NUM==450:
        METRICS_FPATH = METRICS_450_FPATH
        MODEL = model.neural450_net

    # Model and input function
    nn = tf.estimator.Estimator(model_fn=MODEL, model_dir=LOG_PATH)
    train_input_fn = tf.estimator.inputs.numpy_input_fn(
                                            x={"X": X_train}, y=y_train,
                                            batch_size=BATCH_SIZE,
                                            num_epochs=None, shuffle=True)
    eval_input_fn = tf.estimator.inputs.numpy_input_fn(
                                            x={"X": X_eval}, y=y_eval,
                                            num_epochs=1, shuffle=False)

    # Train and evaluate model
    accuracy = []; precision = []; recall = []
    for epoch in range(EPOCH_NUM):
        nn.train(input_fn=train_input_fn, steps=100)
        eval_results = nn.evaluate(input_fn=eval_input_fn)
        accuracy.append(eval_results['accuracy'])
        precision.append(eval_results['precision'])
        recall.append(eval_results['recall'])

    # Save training metrics
    save_dict = {"accuracy":accuracy, "precision":precision, "recall":recall}
    savemat(METRICS_FPATH, save_dict)


#-------------------------------------------------------------------------------
#   Run the application
#-------------------------------------------------------------------------------
tf.app.run()
```

Chương trình kiểm tra ảnh, sau khi đã huấn luyện mạng Nơ-ron học

```
#-------------------------------------------------------------------------------
#   Import library
#-------------------------------------------------------------------------------
from utils import model, image, basic_ft
import lib300
from multiprocessing import Pool, cpu_count
import numpy as np
import tensorflow as tf


#-------------------------------------------------------------------------------
#   Define
#-------------------------------------------------------------------------------
# Files contain testing image paths
AU_SRC_FILE = "label-data/au-files-2-testing.txt"
TP_SRC_FILE = "label-data/tp-files-2-testing.txt"

# Folders contain images
IN_AU_PREFIX = "CASIA-v2/Au/"
IN_TP_PREFIX = "CASIA-v2/Tp/"

# Folder contains training information
LOG_PATH = "log/nn/"

# File contains some configurations
CONFIG_FPATH = "dataset/config-params.mat"

# Sliding parameters
K = 32
S = 16
THRESHOLD = 0.9
SURROUND = 8


#-------------------------------------------------------------------------------
#   Main execution
#-------------------------------------------------------------------------------
# Create parallel pool
pool = Pool(processes=cpu_count())

# Select model and load configurations
MODEL = model.neural300_net
nn = tf.estimator.Estimator(model_fn=MODEL, model_dir=LOG_PATH)
config_params = basic_ft.load(CONFIG_FPATH)

# Select and read image to test
fnames_au = image.read_list_fnames(AU_SRC_FILE)
fname = fnames_au[0]
print('\n', fname)
fpath = IN_AU_PREFIX + fname
img = image.imgRGB(fpath)

# Sliding window to take out patches
coords = image.slide2d(img.shape[:2], K, S)
patches = image.patches(img, coords, K)

# Extract features
feature = lib300.extract_all(patches, pool=pool)
feature_norm = lib300.normalize(feature, config_params)

# Predict raw labels
input_fn = tf.estimator.inputs.numpy_input_fn(
            x={"X": feature_norm}, num_epochs=1, shuffle=False)
results = nn.predict(input_fn=input_fn, predict_keys=["classes"])
```

```
65. labels = np.array([result["classes"] for result in results])
66.
67. # Post-process
68. labels_new = image.post_process(labels, coords, pool=pool,
69.                                  sur=SURROUND, thres=THRESHOLD)
70. last_mark = image.fusion(labels_new)
71.
72. # Print the result
73. print("Label = %d" % last_mark)
```