Một lược đồ mới phát hiện ảnh giả mạo dựa trên luật Benford

A Novel Scheme for Detecting Image Forgeries based on Benford Law

Nguyễn Hiếu Cường, Võ Đức Thắng

Abstract: *Digital image tampering is becoming popular and might cause serious consequences on different areas. Thus, detection of image forgeries is an urgent need. There are various forgery types, which can be exposed by different forensic techniques. In this paper, we propose a new method based on Benford law, also known as the first-digit law, and the SVM classification in order to identify double JPEG compressed images and Gaussian noise added images. Experiments on large-scale image data sets show that the proposed scheme is reliable and it can achieve a high forgery detection capability, with a detection rate is about 90% or higher.*

Keywords: *image forensics, Benford law, SVM, double JPEG compression*

I. GIỚI THIỆU

Ngày nay, cùng với sự phát triển của khoa học kỹ thuật và mạng Internet, ảnh số dễ dàng được thu nhận, lưu trữ, chỉnh sửa và trao đổi. So với ảnh truyền thống dùng phim, ảnh số có nhiều ưu điểm, đặc biệt là dễ dàng chỉnh sửa. Việc hiệu chỉnh ảnh có thể chỉ nhằm mục đích tạo ra những bức ảnh đẹp hơn hoặc có tính nghệ thuật cao hơn. Tuy nhiên, chỉnh sửa ảnh cũng có thể bị lợi dụng để giả mạo và thay đổi nội dung của ảnh với những ý đồ xấu. Bằng một số công cụ xử lý ảnh thông dụng hiện nay, như Photoshop, GIMP, ImageMagick… thì sẽ không khó để thực hiện nhiều thủ thuật xử lý nhằm thay đổi nội dung của ảnh mà không để lại những dấu vết có thể nhận biết được.

Một bức ảnh chỉ có thể được sử dụng để minh chứng cho một sự thật nào đó nếu ảnh đó được xác định là ảnh thật, không bị chỉnh sửa nội dung. Tuy nhiên, khi công bố một bức ảnh đã qua chỉnh sửa, người ta thường chỉ đưa ra bức ảnh sau cùng, chứ không đưa ra ảnh gốc. Do vậy, cần phải có phương pháp đáng tin cậy để xác định một bức ảnh nào đó đã bị biến đổi hay chưa mà không cần có ảnh gốc để đối chiếu. Có rất nhiều cách giả mạo ảnh, do đó cũng cần có nhiều phương pháp khác nhau để phát hiện những sự giả mạo đó [1].

Một phương pháp khá phổ biến trong xác thực ảnh (cũng như các dữ liệu đa phương tiện nói chung) là sử dụng thủy vân số (digital watermarking). Với phương pháp này, một dấu thủy vân được nhúng vào trong ảnh sao cho không tác động nhiều đến chất lượng ảnh (ít nhất là bằng mắt thường không thể nhận biết sự khác biệt giữa ảnh ban đầu và ảnh đã được nhúng thủy vân). Nếu ảnh không bị sửa đổi thì dấu thủy vân vẫn nguyên vẹn khi trích ra, ngược lại, dấu thủy vân sẽ bị biến đổi khác với dấu nhúng ban đầu. Tuy nhiên, trong thực tế, hầu hết các máy ảnh đều không được trang bị chức năng nhúng và trích thủy vân nên phạm vi ứng dụng thủy vân có nhiều hạn chế. Các kỹ thuật thủy vân hiện nay chủ yếu được sử dụng cho mục đích bảo vệ bản quyền các sản phẩm số.

Phương pháp phát hiện ảnh giả mạo (image forensics) có thể hoạt động mà không cần ảnh gốc để đối chiếu và không cần nhúng trước bất kỳ dấu thủy vân nào vào ảnh. Như vậy, nếu coi phương pháp thủy vân là chủ động (cần nhúng trước dấu thủy vân vào ảnh) thì phát hiện ảnh giả mạo là phương pháp bị động. Các kỹ thuật phát hiện giả mạo thường dựa trên quan điểm rằng bất kỳ sự giả mạo nào trên ảnh cũng

tác động vào các đặc tính vốn có của ảnh và để lại những dấu hiệu có thể nhận biết được. Việc tìm ra các dấu hiệu bất thường trên là cơ sở để xác định một bức ảnh đã bị can thiệp, sửa đổi hay chưa.

Ảnh số có thể được tổ chức và lưu trữ dưới nhiều định dạng khác nhau, trong đó định dạng JPEG (Joint Photographic Expert Group) là phổ biến nhất hiện nay. Thuật toán nén ảnh JPEG dựa trên phép biến đổi Cosine rời rạc (Discrete Cosine Transform – DCT), được hỗ trợ bởi rất nhiều ứng dụng và thiết bị. Để thực hiện các thao tác sửa đổi trên một ảnh JPEG, cần thực hiện theo một số bước:

(1) tải ảnh JPEG lên một phần mềm xử lý,

(2) sửa đổi ảnh và

(3) lưu ảnh đó lại dưới định dạng JPEG.

Như vậy, bức ảnh nếu bị sửa đổi thì đã được nén JPEG hai lần, hay còn gọi là nén đúp JPEG (double JPEG compression). Nói cách khác, một ảnh nén đúp JPEG thì nhiều khả năng ảnh đó đã bị sửa đổi, nên phát hiện ảnh nén đúp JPEG là một trong những hướng quan trọng để phát hiện ảnh giả mạo.

Với mục đích phát hiện ảnh nén đúp JPEG, một số kỹ thuật đã được đề xuất. Các tác giả trong [2] và [3] đã phát hiện rằng khi tỷ lệ nén của hai lần nén JPEG khác nhau, các dấu hiệu tuần hoàn sẽ xuất hiện trong biểu đồ tần suất (histogram) các hệ số DCT của ảnh nén đúp JPEG, trong khi các dấu hiệu này không xuất hiện ở ảnh chỉ nén JPEG một lần. Các dấu hiệu tuần hoàn đó có thể nhận biết được một cách trực quan qua quan sát phổ Fourier (Fourier spectrum) khi biến đổi ảnh sang miền không gian. Tuy nhiên, kỹ thuật này chỉ hoạt động tốt khi chất lượng nén JPEG lần thứ hai cao hơn lần nén thứ nhất. Ngược lại, khi chất lượng nén JPEG lần thứ hai thấp hơn chất lượng nén lần đầu thì tỷ lệ phát hiện giả mạo rất thấp.

Dựa vào ý tưởng trong [2] và [3], He và các đồng sự [4] đã đề xuất một kỹ thuật phát hiện ảnh ghép JPEG. Dựa trên đặc tính của kỹ thuật nén JPEG, Farid [5] đã đưa ra một phương pháp để tìm được sự không tương thích của các khối ảnh khi ghép hai ảnh JPEG với nhau. Tuy vậy, kỹ thuật của Farid chỉ phù hợp khi phần được ghép vào một bức ảnh có chất lượng nén JPEG thấp hơn những phần còn lại của bức ảnh đó. Chen và các đồng sự [6] đã đề xuất một lược đồ dựa trên phương pháp học máy để phát hiện ảnh nén đúp JPEG.

Luật Benford [7] bắt đầu được nghiên cứu và ứng dụng trong phát hiện ảnh giả mạo từ công trình của Fu và các đồng sự [8]. Một số công trình khác đã cụ thể hóa một số ý tưởng của [8], ví dụ [9]. Milani và các đồng sự [10] đã sử dụng luật Benford để xác định các ảnh JPEG được nén nhiều lần.

Trong bài báo này, chúng tôi trình bày một lược đồ hoàn chỉnh sử dụng các đặc trưng Benford kết hợp với kỹ thuật học máy SVM (Support Vector Machine) để phát hiện nhiều loại ảnh giả mạo khác nhau. Trước hết, chúng tôi ứng dụng lược đồ trên để phát hiện ảnh giả mạo kiểu nén đúp JPEG. Kết quả thử nghiệm phương pháp của chúng tôi đề xuất sẽ được so sánh với một số phương pháp đang được sử dụng rộng rãi, như thống kê tần suất [3] và sử dụng phương pháp học máy SVM [6].

Chúng tôi cũng ứng dụng lược đồ đề xuất này để phân lớp giữa ảnh gốc JPEG và ảnh JPEG đã bị thêm nhiễu. Việc thêm nhiễu là một kỹ thuật tấn công thường được sử dụng trong các quá trình làm giả ảnh. Mục đích của việc thêm nhiễu là để che giấu những dấu hiệu của việc làm giả trước đó, nhằm đánh lừa hoặc vô hiệu hóa các thuật toán phát hiện ảnh giả mạo. Do đó, một ảnh bị thêm nhiễu bất thường cũng có nhiều khả năng là một ảnh giả. Theo hiểu biết của chúng tôi, cho đến nay chưa có một công trình nào ứng dụng luật Benford để phát hiện một ảnh đã bị thêm nhiễu. Trong bài báo này, chúng tôi lần đầu tiên sử dụng lược đồ dựa trên luật Benford để đánh giá một ảnh có bị thêm nhiễu Gauss hay không.

Trong những phần tiếp theo, trước hết chúng tôi giới thiệu một số khái niệm cơ bản sẽ được sử dụng trong bài báo, đó là nén ảnh JPEG và luật Benford. Lược đồ áp dụng luật Benford để phát hiện ảnh JPEG

giả mạo được trình bày trong phần III. Quy trình và các kết quả thử nghiệm được mô tả chi tiết hơn trong phần IV. Kết quả được thử nghiệm trên các tập lớn dữ liệu ảnh giả mạo các loại cho thấy ứng dụng luật Benford là một hướng tiếp cận hiệu quả để phát hiện ảnh giả mạo. Cuối cùng là kết luận và tài liệu tham khảo.

II. MỘT SỐ KHÁI NIỆM CƠ SỞ

II.1. Nén ảnh JPEG

Nén ảnh là một phương pháp hữu hiệu để giảm kích thước lưu trữ nhưng vẫn đảm bảo được chất lượng hình ảnh ở mức cho phép. Thuật toán nén ảnh JPEG đang được sử dụng phổ biến nhất hiện nay do có thể giảm đáng kể dung lượng lưu trữ trong khi vẫn đảm bảo tốt chất lượng ảnh. Tùy theo nhu cầu sử dụng mà chúng ta có thể nén ảnh JPEG với các tỷ lệ nén khác nhau.

Trong quy trình nén ảnh JPEG, đầu tiên ảnh được chuyển đổi sang không gian màu YCrCb, sau đó mỗi kênh Y, Cr, Cb sẽ được xử lý riêng rẽ theo cách tương tự nhau. Ảnh đa mức xám (grayscale) được xử lý tương tự như thực hiện trên từng kênh màu ở trên, gồm các bước chính được mô tả như sau [11]:

Bước 1: Ảnh nguồn được chia thành các khối 8×8 không giao nhau.

Bước 2: Thực hiện biến đổi DCT cho mỗi khối ảnh. Các giá trị của khối sau khi biến đổi gọi là các hệ số DCT, trong đó hệ số đầu tiên (ở vị trí hàng 1, cột 1 của mỗi khối) gọi là hệ số DC, các hệ số còn lại trong khối gọi là các hệ số AC. Do đặc trưng tập trung năng lượng của phép biến đổi DCT, giá trị của hệ số DC thường lớn hơn rất nhiều so với giá trị của các hệ số AC.

Bước 3: Lượng tử hóa các hệ số DCT của từng khối bằng cách lấy phần nguyên của phép chia từng hệ số của khối DCT với hệ số tương ứng (cùng vị trí) của ma trận lượng tử 8×8. Các giá trị sau bước lượng tử gọi là các hệ số DCT lượng tử.

Bước 4: Mã hóa entropy để tạo thành tệp ảnh JPEG.

Khi cần tái hiện ảnh JPEG, các bước thực hiện theo quy trình ngược lại, gồm các bước chính là giải nén tệp ảnh JPEG và biến đổi DCT ngược (IDCT).

II.2. Luật Benford

Luật Benford là một định luật thực nghiệm được phát hiện lần đầu bởi S. Newcomb năm 1881, sau đó được làm rõ và bắt đầu ứng dụng bởi F. Benford vào năm 1938 [7]. Luật này chỉ ra rằng các chữ số đầu tiên của một tập số liệu lớn trong tự nhiên thường có phân bố theo một qui luật. Cụ thể, luật Benford chỉ ra rằng xác suất phân bố của các chữ số thứ nhất x trong một tập lớn số liệu tự nhiên là theo dạng logarith như sau:

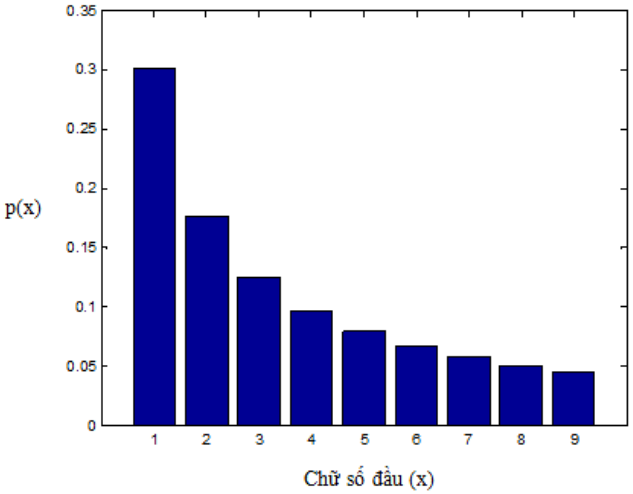
$$p(x) = \log_{10}(1 + 1/x), \text{ với } x = 1, 2, \ldots, 9,$$

trong đó p(x) là xác suất phân bố của chữ số x. Tỷ lệ phân bố các chữ số đầu theo luật Benford có thể thấy như trong Hình 1.

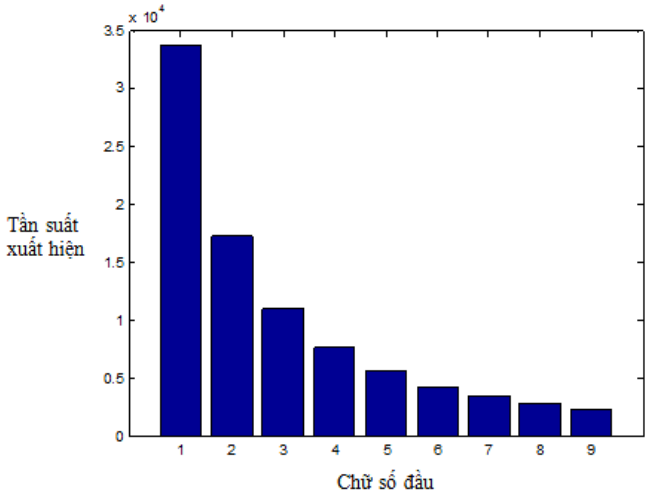
Điều kiện để áp dụng luật Benford là:

(1) tập dữ liệu mẫu phải đủ lớn và

(2) chỉ áp dụng cho những loại dữ liệu có nguồn gốc tự nhiên.



Hình 1. Phân bố các chữ số đầu theo luật Benford



Hình 2. Tần suất xuất hiện các chữ số đầu của các hệ số DCT lượng tử của một ảnh JPEG

Dữ liệu của một bức ảnh chưa qua chỉnh sửa có thể được coi là có nguồn gốc tự nhiên. Nếu xét một bức ảnh JPEG có kích cỡ bình thường trong thực tế thì số hệ số DCT lượng tử là khá lớn nên có thể áp dụng luật Benford. Ví dụ, với một ảnh JPEG kích cỡ 512×318, số hệ số DCT lượng tử lên đến hàng chục ngàn. Tần suất xuất hiện của chữ số đầu của các hệ số DCT lượng tử (chỉ tính riêng các hệ số AC khác 0) của một bức ảnh như vậy có thể được mô tả như trong Hình 2.

II.3. Phương pháp học máy SVM

SVM là một phương pháp phân lớp dựa trên lý thuyết thống kê, được đề xuất bởi Vapnik [12]. Đây là phương pháp cho phép phân lớp dữ liệu bằng cách sử dụng các hàm tuyến tính trên không gian đặc trưng nhiều chiều, dựa vào lý thuyết tối ưu và lý thuyết thống kê. SVM được xem như là một trong các phương pháp phân lớp tinh vi và hiệu quả nhất hiện nay.

Trong phương pháp SVM, dữ liệu ban đầu tương ứng với không gian mẫu đầu vào sẽ được chuyển đổi (ánh xạ) thành một không gian đặc trưng riêng. Tại đây, có thể xác định một siêu phẳng thực hiện phân chia tối ưu các mẫu thành hai miền xác định. Các hàm chuyển đổi đa dạng của SVM cho phép tạo không gian chuyển đổi một cách linh hoạt cho các dữ liệu đầu vào.

Cho trước một tập huấn luyện bao gồm những thuộc tính và nhãn cho từng đối tượng. Các đối tượng được biểu hiện là từng điểm trong không gian vector. Thuật toán SVM sẽ cố gắng xác định một siêu phẳng quyết định tốt nhất sao cho có thể phân chia các điểm trong không gian vector này thành hai lớp riêng biệt. Chất lượng phân chia của siêu phẳng này quyết định bởi khoảng cách (biên) của điểm dữ liệu gần nhất của mỗi lớp đến mặt phẳng phân chia này. Do đó, khoảng cách biên càng lớn thì mặt phẳng quyết định càng tốt, do đó khả năng phân loại sẽ càng chính xác.

Thông thường dữ liệu đầu vào không dễ dàng phân chia trực tiếp thành hai lớp phân biệt một cách tối ưu nên cần sử dụng các hàm nhân (kernel function) để thực hiện chuyển đổi không gian ban đầu thành một không gian vector khác (không gian đặc trưng) sao cho có thể phân chia được thành hai lớp. Khi đó, số thuộc tính của từng đối tượng trong không gian mới sẽ tăng lên, thời gian tính toán tương ứng cũng tăng theo nhưng đã thỏa mãn được yêu cầu là phân loại được dữ liệu. Tuy nhiên, để quá trình phân loại thực hiện một cách chính xác nhất thì cần quá trình điều chỉnh hàm nhân với các tham số đi kèm.

Việc phân loại dùng SVM gồm các bước chính:

Bước 1: SVM yêu cầu dữ liệu đầu vào dùng để huấn luyện phải được mô tả như là các vector số thực nên cần có bước tiền xử lý để biến đổi dữ liệu cho phù hợp với quá trình tính toán, tránh các số có giá trị quá lớn. Dữ liệu đó nên chuyển về đoạn [-1, 1] hoặc [0, 1].

Bước 2: Do SVM có thể dùng các hàm nhân khác nhau nên việc xác định đúng loại hàm nhân cần dùng cho một bài toán cụ thể có thể giúp đạt độ chính xác cao hơn.

Bước 3: Thực hiện việc kiểm tra chéo (cross validation) để xác định các tham số tối ưu.

Bước 4: Có thể thử nghiệm trên các tập dữ liệu dùng kiểm tra để xác định độ chính xác.

III. LƯỢC ĐỒ PHÁT HIỆN ẢNH GIẢ MẠO

Ảnh thật là ảnh ghi nhận các đối tượng từ thế giới thực với những khoảng biến đổi liên tục về không gian và màu sắc. Giá trị các điểm ảnh là những giá trị từ thực tế, không có sự can thiệp của con người và mỗi bức ảnh thông thường có số lượng điểm ảnh tương đối nhiều. Đây là những điều kiện cần thiết để có thể áp dụng luật Benford.

Khi một bức ảnh bị làm giả thì các giá trị trong ảnh đã bị can thiệp một phần, do đó có thể làm cho các thông số thống kê của ảnh không còn tuân theo luật Benford. Vì thế đặc tính này sẽ là yếu tố để xác định xem một bức ảnh có bị chỉnh sửa không. Trong phần này, chúng tôi trình bày giải thuật kết hợp luật Benford và phân lớp dữ liệu sử dụng SVM để phát hiện ảnh nén đúp JPEG.

Đầu vào của thuật toán là ảnh JPEG và đầu ra của thuật toán là một giá trị để có thể phân biệt ảnh thật (ảnh nén JPEG) và ảnh giả (ví dụ ảnh nén đúp JPEG). Các bước chính của thuật toán được tiến hành như mô tả dưới đây.

Bước 1: Mỗi ảnh JPEG sẽ được giải nén để lấy các hệ số DCT lượng tử. Để làm được điều này, trong chương trình thử nghiệm chúng tôi sử dụng một công cụ miễn phí là JPEGToolbox của Phil Sallee [13].

Bước 2: Tại mỗi khối, xác định tần suất xuất hiện của các chữ số đầu khác 0 của các hệ số AC lượng tử. Ví dụ, nếu dòng đầu tiên trong một khối DCT lượng tử có giá trị là [152 4 23 12 6 3 0 0] thì các chữ số đầu tương ứng sẽ là [* 4 2 1 6 3 * *], trong đó dấu * để thể hiện những số không được dùng (giá trị 152 là hệ số DC và giá trị 0 không được sử dụng). Thực hiện tương tự cho toàn bộ các khối ảnh để tính các giá trị đặc trưng Benford của ảnh.

Đặc trưng Benford của ảnh sau khi được trích xuất sẽ đồng thời được gán nhãn để hỗ trợ phân lớp bằng thuật toán học máy SVM. Thông tin về đặc trưng Benford của ảnh sẽ bao gồm một vector 10 phần tử, trong đó có 9 phần tử là xác định tỉ lệ chênh lệch giữa phân bố thực tế và phân bố theo luật Benford của các hệ số AC lượng tử của ảnh JPEG và phần tử còn lại xác định nhãn tương ứng.

Cụ thể, đặc trưng Benford của mỗi ảnh JPEG được tính là một bộ (d_0, d_1, d_2, …, d_9), trong đó d_0 là nhãn và d_i (i = 1, 2 … 9) được tính theo công thức sau:

$$d_i = |\log_{10}(1+1/i) - t_i|,$$

với t_i là xác suất xuất hiện của chữ số đầu i trong tập các hệ số AC lượng tử khác 0 của ảnh.

Bước 3: Trích các giá trị đặc trưng Benford (như mô tả trong Bước 2) của một tập lớn các ảnh thật và một tập lớn ảnh giả (ảnh nén đúp JPEG), để thực hiện huấn luyện bằng phương pháp SVM.

Bước 4: Sau khi đã được huấn luyện, chương trình có thể được sử dụng để xác định một ảnh JPEG nào đó là ảnh thật hay giả.

Lược đồ trên được áp dụng trước hết để phát hiện ảnh nén đúp JPEG. Tiếp sau đó, lược đồ cũng được chúng tôi áp dụng (huấn luyện và kiểm tra) theo cách thức hoàn toàn tương tự để phát hiện ảnh JPEG bị thêm nhiễu Gauss. Trong trường hợp này, ảnh thật là ảnh nén JPEG (nén một lần) và ảnh giả là ảnh JPEG được thêm nhiễu Gauss với các mức độ khác nhau. Trong các quá trình huấn luyện và kiểm tra, thay vì sử dụng các ảnh giả là ảnh nén đúp JPEG, chúng tôi dùng ảnh giả là các ảnh JPEG đã được thêm nhiễu Gauss với các mức độ khác nhau. Việc chuẩn bị dữ liệu và kết quả thử nghiệm trên các tập dữ liệu lớn được trình bày ở phần tiếp theo.

IV. KẾT QUẢ THỬ NGHIỆM

IV.1. Dữ liệu và phương pháp thử nghiệm

Để chuẩn bị dữ liệu thử nghiệm, chúng tôi sử dụng một tập gồm 1338 ảnh màu không nén, chưa từng bị sửa đổi, có kích thước 512×318 hoặc 318×512 trong cơ sở dữ liệu UCID (Uncompressed Color Image Database) [14]. Đây là một cơ sở dữ liệu ảnh chuẩn, miễn phí, được sử dụng phổ biến trong nhiều nghiên cứu về xử lý ảnh. Tiếp theo, các ảnh này được nén JPEG với hệ số chất lượng (QF – Quality Factor) lần lượt là 50, 55, 60, 65, 70, 75, 80, 85, 90 và 95 để tạo

ra các ảnh gốc JPEG (ảnh thật), bao gồm trong 10 tập, mỗi tập có 1338 ảnh. Để tạo các ảnh giả nén đúp JPEG, chúng tôi lấy tập ảnh JPEG với QF = 50 và nén JPEG một lần nữa, lần lượt với chất lượng nén ảnh từ 55% đến 100% để tạo ra 10 tập ảnh ảnh (nén đúp JPEG), mỗi tập có 1338 ảnh.

Tập ảnh dùng để huấn luyện là tập ảnh bao gồm các ảnh thật và ảnh giả. Tập ảnh dùng để kiểm tra là tập ảnh được chọn ngẫu nhiên và không trùng lặp với các ảnh đã được dùng trong quá trình huấn luyện để đảm bảo tính khách quan khi kiểm tra.

IV.2. Thử nghiệm với ảnh nén đúp JPEG

Với mỗi tập ảnh (tương ứng với một tỷ lệ nén), chúng tôi chọn ngẫu nhiên 1200 ảnh để huấn luyện và 138 ảnh còn lại dùng để kiểm tra. Sau quá trình huấn luyện, chúng tôi kiểm tra với những tập ảnh JPEG với tỷ lệ nén khác nhau. Cụ thể, dữ liệu kiểm tra gồm 10 tập ảnh thật (ảnh JPEG nén một lần), 10 tập ảnh giả (ảnh nén đúp JPEG) và 10 tập ảnh hỗn hợp (với tỷ lệ một nửa ảnh thật và một nửa ảnh giả), mỗi tập có 138 ảnh. Như vậy, tổng số ảnh JPEG dùng để huấn luyện là 36000 ảnh và dùng để kiểm tra là hơn 4000 ảnh.

Trong quy trình đánh giá, trước hết chương trình (sau khi đã huấn luyện) được sử dụng để kiểm tra trên các tập gồm toàn các ảnh giả (ảnh nén đúp JPEG). Tiếp đó chúng tôi sử dụng chương trình này để kiểm tra các thư mục gồm toàn ảnh thật (ảnh nén JPEG một lần). Cuối cùng, chúng tôi kiểm tra với các tập ảnh hỗn hợp, gồm cả ảnh thật và ảnh giả (mỗi tập gồm 69 ảnh thật và 69 ảnh giả).

Chúng tôi sử dụng hàm fitcsvm trong Matlab để thực hiện việc huấn luyện cho dữ liệu mẫu theo giải thuật SVM với hàm nhân là Radial Basis Function (RBF). Để quá trình huấn luyện chính xác hơn, chúng tôi sử dụng thêm tính năng kiểm tra chéo kết quả huấn luyện nhằm xác định tham số huấn luyện tối ưu.

Điều trên được thực hiện thông qua các bước sau:

Bước 1: Thiết lập phân vùng giới hạn việc kiểm tra chéo. Nếu không có bước này, việc kiểm tra sẽ được thực hiện ngẫu nhiên, do đó, cần xác định chính xác khu vực kiểm tra để có kết quả tốt hơn.

Bước 2: Ngoài ra, để thực hiện cần phải có thêm hai tham số truyền vào là rbf_sigma và boxconstraint.

Bước 3: Có thể tìm kiếm các thông số tốt nhất cho cặp giá trị rbf_sigma và boxconstraint thông qua hàm fminsearch với các thông số mặc định.

Bước 4: Khi đó, giá trị tốt nhất cho z = [rbf_sigma, boxconstraint] sẽ là z = exp(searchmin). Tuy nhiên, đây chỉ là kết quả tối ưu nhất thời, không phải là tối ưu cho toàn bộ mẫu huấn luyện. Do đó, chúng ta cần thực hiện lặp lại nhiều lần và chọn một giá trị tốt nhất để dùng cho toàn bộ mẫu huấn luyện. Kết quả cũng được coi là đáng tin cậy hơn nếu giữa các lượt thử nghiệm không quá khác biệt lớn. Trong bài báo này, chúng tôi thực hiện hai lượt thử nghiệm theo lược đồ và bộ dữ liệu đã trình bày ở trên. Chúng tôi đã sử dụng hàm predict của Matlab để xem xét dạng ảnh của tập mẫu có ứng với dữ liệu huấn luyện không. Kết quả trả về là một vector chứa nhãn tương ứng của các mẫu đã được kiểm tra.

Kết quả thử nghiệm trong Bảng 1 và Bảng 2 cho thấy, khi thực hiện kiểm tra trên các tập gồm toàn ảnh giả, tỷ lệ phát hiện được ảnh giả là rất cao, khoảng 90% trở lên và trong quá nửa số trường hợp kiểm tra, tỷ lệ này là từ 99% đến 100%. Không chỉ kiểm tra trên các tập gồm toàn ảnh giả như trong một số nghiên cứu thường thực hiện, chúng tôi đã thử nghiệm trên cả tập ảnh hỗn hợp (gồm 50% số ảnh thật và 50% số ảnh giả). Sau khi kiểm tra trên các tập hỗn hợp này cho thấy kết quả không khác nhiều so với khi thực hiện trên các tập toàn ảnh giả, với tỷ lệ tối thiểu là 85% và trong nhiều trường hợp vẫn có thể đạt tỷ lệ phát hiện giả mạo là 100%. Điều đó thể hiện rằng khả năng phân loại của chương trình là tốt. Ngoài ra, khi kiểm tra các tập chứa toàn ảnh thật, chúng tôi nhận thấy, kết quả trong tất cả các trường hợp, tỷ lệ ảnh giả phát hiện được đều là 0%, chứng tỏ không có sự phát hiện nhầm một ảnh thật là ảnh giả. Đây là một chỉ số quan trọng, vì nếu một phương pháp có tỷ lệ phát hiện

giả mạo cao nhưng tỷ lệ phát hiện nhầm cũng cao thì không đủ tin cậy để ứng dụng trong thực tế. Với những thử nghiệm trong [3, 6], các tác giả đã không đề cập đến tỷ lệ phát hiện nhầm này.

Để so sánh với các thuật toán của [3, 6], chúng tôi sử dụng cùng một bộ dữ liệu ảnh như trong thử nghiệm thuật toán đề xuất ở trên. Kết quả thử nghiệm các thuật toán của [3] và [6] được trình bày trong Bảng 3. Lưu ý rằng các kết quả này chỉ được thực hiện khi kiểm tra trên các tập dữ liệu gồm toàn ảnh giả.

Bảng 1. Tỷ lệ phát hiện ảnh giả mạo nén đúp JPEG (Lượt thử nghiệm thứ nhất) [%]

Chất lượng nén ảnh (QF)	Kiểm tra tập toàn ảnh giả	Kiểm tra tập ảnh hỗn hợp
55	91	89
60	96	96
65	100	100
70	99	97
75	95	97
80	100	100
85	99	98
90	99	97
95	99	98
100	100	100

Bảng 2. Tỷ lệ phát hiện ảnh giả mạo nén đúp JPEG (Lượt thử nghiệm thứ hai) [%]

Chất lượng nén ảnh (QF)	Kiểm tra tập toàn ảnh giả	Kiểm tra tập ảnh hỗn hợp
55	88	85
60	94	85
65	100	100
70	98	100
75	97	99
80	100	100
85	89	97
90	99	98
95	100	100
100	99	97

Bảng 3. Tỷ lệ phát hiện ảnh giả mạo nén đúp JPEG của lược đồ Popescu và lược đồ Chen [%]

Chất lượng nén ảnh (QF)	Lược đồ của Popescu [3]	Lược đồ của Chen [6]
55	100	100
60	99	100
65	98	100
70	100	100
75	97	100
80	93	100
85	86	100
90	85	99
95	88	94
100	84	92

IV.3. Thử nghiệm với ảnh thêm nhiễu Gauss

Trong phần này, chúng tôi sử dụng lược đồ tương tự như trong IV.2 để phát hiện một ảnh có bị thêm nhiễu hay không. Ảnh thật bao gồm 10 tập ảnh JPEG với chất lượng nén từ 50% đến 95%, mỗi tập gồm 1338 ảnh (xem mục IV.1). Ảnh giả được tạo bằng cách thêm nhiễu Gauss với cường độ khác nhau từ 5% đến 10% vào các ảnh JPEG. Kết quả chúng tôi có 10 tập ảnh giả (ảnh JPEG bị thêm nhiễu), mỗi tập gồm 1338 ảnh. Các tập ảnh để huấn luyện bao gồm trong 10 tập ảnh thật, 10 tập ảnh giả và 10 tập ảnh hỗn hợp (gồm 50% số ảnh JPEG thật và 50% số ảnh JPEG giả), mỗi tập ảnh gồm 1200 ảnh. Dữ liệu để kiểm tra cũng gồm có 30 tập như trên, trong đó mỗi tập gồm 138 ảnh. Như vậy, tương tự như thử nghiệm trong IV.2, chúng tôi đã sử dụng 36000 ảnh để huấn luyện và hơn 4000 ảnh để kiểm tra. Để cho kết quả chính xác hơn, chúng tôi cũng thực hiện hai lượt kiểm tra như trong IV.2. Kết quả thử nghiệm được trình bày trong Bảng 4 và Bảng 5. Chúng tôi lần đầu tiên ứng dụng luật Benford để phát hiện ảnh bị thêm nhiễu và nhận thấy, khả năng phát hiện giả mạo với loại ảnh này cũng rất tốt, với tỷ lệ phát hiện khoảng từ 90% đến 100%. Khi thử nghiệm trên tập toàn ảnh thật, tỷ lệ ảnh giả phát hiện được là 0%, tức là không có sự phát hiện nhầm một ảnh thật là ảnh giả.

Bảng 4. Tỷ lệ phát hiện ảnh giả mạo thêm nhiễu Gauss (Lượt thử nghiệm thứ nhất) [%]

Chất lượng nén ảnh (QF)	Kiểm tra tập toàn ảnh giả	Kiểm tra tập ảnh hỗn hợp
50	93	90
55	95	91
60	94	94
65	94	96
70	100	100
75	98	97
80	96	96
85	96	95
90	94	97
95	100	100

Bảng 5. Tỷ lệ phát hiện ảnh giả mạo thêm nhiễu Gauss (Lượt thử nghiệm thứ hai) [%]

Chất lượng nén ảnh (QF)	Kiểm tra tập toàn ảnh giả	Kiểm tra tập ảnh hỗn hợp
50	96	94
55	93	93
60	93	91
65	93	97
70	97	98
75	100	100
80	97	96
85	97	96
90	97	97
95	100	98

V. KẾT LUẬN

Bài báo này đã trình bày một lược đồ hoàn chỉnh ứng dụng luật Benford (hay còn gọi là luật chữ số thứ nhất) để phát hiện ảnh giả mạo dựa trên việc xác định ảnh đó có các dấu hiện nén đúp JPEG hoặc có bị thêm nhiễu Gauss hay không.

Bước đầu tiên, dựa trên các hệ số DCT lượng tử của ảnh JPEG, chúng tôi thiết kế các đặc trưng Benford của ảnh. Tiếp đó, chúng tôi trích xuất các đặc trưng Benford của các ảnh trong một tập lớn ảnh JPEG các loại để tiến hành huấn luyện với SVM nhằm mục đích phân lớp giữa ảnh JPEG gốc và ảnh JPEG có chỉnh sửa (nén đúp JPEG hoặc thêm nhiễu Gauss). Kiểm tra trên các tập dữ liệu lớn gồm hàng nghìn ảnh JPEG đã cho kết quả phát hiện giả mạo khoảng từ 90% đến 100% và không có trường hợp nào phát hiện nhầm ảnh thật là ảnh giả.

Trong thực tế, không phải ảnh số nào có dấu hiệu nén đúp JPEG hoặc có nhiễu Gauss cũng là ảnh giả. Chẳng hạn, ảnh JPEG có thể được nén lại một lần nữa với tỷ số nén cao hơn chỉ nhằm làm giảm không gian lưu trữ hoặc ảnh có thể bị nhiễu trong quá trình thu nhận và truyền tải ảnh. Tuy nhiên, ảnh JPEG giả mạo thì thường có các dấu hiệu nén đúp JPEG hoặc có nhiễu bất thường. Do đó, phát hiện các dấu hiệu trên là một biện pháp hữu hiệu hỗ trợ phát hiện ảnh giả mạo. Để nâng cao hiệu quả và độ tin cậy trong phát hiện ảnh giả mạo, chúng ta có thể kết hợp nhiều phương pháp phát hiện khác nhau.

TÀI LIỆU THAM KHẢO

[1] A. PIVA, *"An Overview on image forensics"*, ISRN Signal Process., vol. 2013, pp. 1–22, 2013.

[2] J. LUKAS, J. FRIDRICH, *"Estimation of primary quantization matrix in double compressed JPEG Images",* Proc. Digit. Forensic Res. Work., 2003.

[3] A. POPESCU, *"Statistical tools for digital image forensics"*, PhD Thesis, Dartmouth College, Hanover, NH, USA, 2004.

[4] J. HE, Z. LIN, L. WANG, X. TANG, *"Detecting Doctored J PEG Images Via DCT Coefficient Analysis"*, ECCV LNCS 3953, pp. 423–435, 2006.

[5] H. FARID, *"Exposing Digital Forgeries From JPEG Ghosts"*, IEEE Trans. Inf. Forensics Secur., vol. 4, no. 1, pp. 154–160, Mar. 2009.

[6] C. CHEN, Y. SHI, W. SU, *"A machine learning based scheme for double JPEG compression detection"*, Proc. 19th Int. Conf. Pattern Recognization (ICPR), 2008.

[7] F. BENFORD, *"The law of anamalous numbers"*, Proc. Amer. Phi. Soc., vol. vol. 78, pp. 551–572, 1938.

[8] D. FU, Y.Q. SHI, and W. SU, *"A generalized Benford's law for JPEG coefficients and its applications in image forensics"*, Proc. SPIE. 2007.

[9] N. CƯỜNG, V. THẮNG, *"Ứng dụng luật chữ số thứ nhất phát hiện dấu hiệu giả mạo trên ảnh JPEG"*, Tạp chí Khoa học Giao thông vận tải, 11/2015.

[10] S. MILANI, M. TAGLIASACCHI, S. TUBARO, *"Discriminating multiple JPEG compressions using first digit features"*, APSIPA Trans. on Signal and Information Processing, Vol 3, 2014.

[11] G. WALLACE, *"The JPEG still picture compression standard"*, IEEE Trans. Consum. Electron., pp. 1–17, 1991.

[12] V. VAPNIK, *"Statistical learning theory"*, John Wiley, 1998.

[13] http://dde.binghamton.edu/download/jpeg_toolbox.zip

[14] G. SCHAEFER, M. STICH, *"UCID - An Uncompressed Colour Image Database"*, 2004.

Nhận bài ngày: 27/08/2016

SƠ LƯỢC VỀ TÁC GIẢ

NGUYỄN HIẾU CƯỜNG

Sinh ngày 21/5/1974 tại Hải Dương.

Tốt nghiệp ĐH năm 1995 và Cao học năm 1999 tại trường ĐH Quốc gia Hà Nội.

Nhận học vị Tiến sĩ năm 2013 tại trường ĐH Darmstadt, CHLB Đức.

Hiện là giảng viên khoa CNTT, trường ĐH Giao thông vận tải Hà Nội.

Lĩnh vực quan tâm: Xử lý ảnh, an toàn thông tin.

Email: cuonggt@gmail.com

Điện thoại: 0967886712

VÕ ĐỨC THẮNG

Sinh ngày 6/2/1984 tại TP. Hồ Chí Minh.

Tốt nghiệp ĐH năm 2008 tại trường ĐH Công nghiệp TP. Hồ Chí Minh.

Tốt nghiệp Cao học năm 2015 tại trường ĐH Giao thông vận tải (Phân hiệu tại TP. Hồ Chí Minh).

Hiện làm việc tại công ty VTRADE Việt Nam.

Lĩnh vực quan tâm: Xử lý ảnh, ảo hóa.

Email: ducthang128@gmail.com