

PHÁT HIỆN HÌNH ẢNH KHUÔN MẶT GIẢ MẠO ĐƯỢC CHỤP TỪ MÀN HÌNH LCD

**Người thực hiện: Lê Đức Hiệu
Người giám sát: A. Vũ Trung Kiên**

MỤC LỤC

I. Mô tả dự án.....	3
1.1 Đặt vấn đề.....	3
1.2 Mục đích bài toán.....	3
1.3 Mục tiêu bài toán.....	3
1.4 Giao phẩm.....	8
1.5 Những vấn đề rủi ro.....	8
II. Phương pháp thực hiện.....	9
2.1 Tổng quan.....	9
2.2 Image statistics feature extraction.....	9
2.3 SVM classifier.....	12
III. Đề xuất triển khai.....	13
IV. Tổ chức dự án.....	13
V. Ứng dụng và kết luận.....	14

I.Mô tả dự án

1.1 Đặt vấn đề

Bộ phận RnD đã phát triển một phần mềm nhận diện khuôn mặt phục vụ điểm danh. Phần mềm nhận diện khuôn mặt này phát hiện khuôn mặt thu được từ camera và định danh khuôn mặt đó. Tuy nhiên, người dùng có thể qua mặt phần mềm bằng cách giả mạo khuôn mặt. Cụ thể, người dùng có thể dùng ảnh chụp khuôn mặt của người khác để chấm công hộ, hoặc tự chụp ảnh của mình và đưa trước khung hình camera để phần mềm nhận dạng khuôn mặt giả mạo đó.

Như vậy cần có một giải pháp để phát hiện khuôn mặt thu từ camera có phải là khuôn mặt giả mạo hay không. Vì vậy tôi đề xuất một giải pháp giúp hỗ trợ ý tưởng này để chống lại các loại giả mạo được ghi lại qua màn hình điện thoại, laptop. Giải quyết được vấn đề này sẽ gia tăng tính bảo mật cho phần mềm nhận diện khuôn mặt. Cụ thể phần mềm nhận diện khuôn mặt có thể phát hiện khuôn mặt giả mạo và bỏ qua kết quả nhận dạng

1.2 Mục đích bài toán

Như tên gọi của bài toán, mục đích chính là giúp cho các hệ thống nhận diện khuôn mặt biết được người đang đứng trước camera là thật hay được chụp qua màn hình LCD. Từ đó đạt được mức độ bảo mật cao hơn trong những ứng dụng nổi bật của nhận diện khuôn mặt như: chấm công thông minh, bảo vệ thực thi pháp luật, mở khóa điện thoại

1.3 Mục tiêu

- 3/2/2021 bắt đầu project: Tiến hành survey các cách giải quyết và đưa ra phương pháp tối ưu cuối cùng để thực hiện (ước tính 2 ngày)
- 5/2/2021 lên được pipeline giải quyết các giai đoạn thực hiện project, tiến hành thu thập dữ liệu về khuôn mặt cho bài toán (ước tính 1 ngày)
- 6/2/2021 Tiến hành xử lý bộ dataset celebA-Spoof, lọc và làm sạch dữ liệu chuẩn bị cho quá trình training, hiểu được cấu trúc của dataset (ước tính 1 ngày)
- 7/2/2021 Xây dựng thành công model pipeline, đưa ra các metric đánh giá cho quá trình training.(ước tính 2 ngày)
- 15/2/2021 Tiến hành training model trên tập dataset celebA-Spoof mục tiêu với accuracy > 85%, tiến hành (ước tính 1 ngày)
- 16/2/2021 Phân tích các nguyên nhân gây ra lỗi trên tập test set, đưa ra các biện pháp cải thiện độ chính xác, training lại với mục tiêu đạt accuracy > 90% trên tập test (ước tính 1 ngày)
- 17/2/2021 Thu thập và xử lý dữ liệu thực tế từ camera ip chuẩn bị cho quá trình kiểm định thực tế (ước tính 2 ngày)
- 19/2/2021 Tune lại model trên dữ liệu thực tế với average accuracy > 95%, đánh giá mức độ tin cậy của hệ thống khi hoạt động. (ước tính 1 ngày)

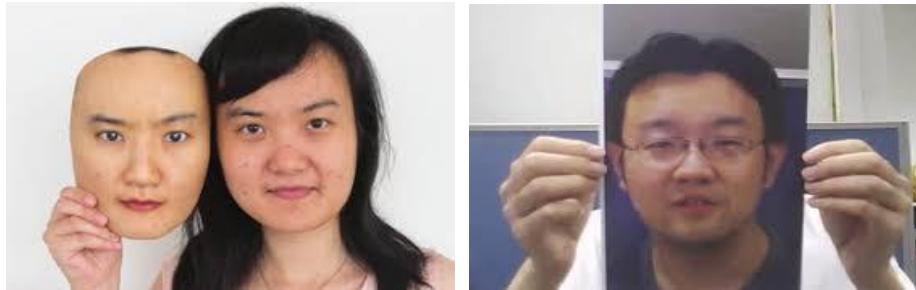
1.4. Giao phẩm

Source code: https://github.com/phuthu19112000/spoof_image_face

1.5. Những vấn đề và rủi ro

- Trong quá trình đào tạo, đánh giá và thử nghiệm trên dữ liệu thực tế , thấy có những vấn đề sau:

1. Vì đây là bài toán phát hiện khuôn mặt giả mạo trên màn hình LCD cho nên khi một người thực hiện các loại giả mạo khác như đeo mặt nạ, chụp ảnh trên giấy thì trình phân loại SVM sẽ bị nhận dạng sai. Ví dụ về loại giả mạo trên mặt nạ.



Hình 4: Bên phải là hình ảnh giả mạo trên mặt nạ, trái là giả mạo trên giấy

2. Đối với những loại giả mạo trên LCD hạn chế là bức ảnh phải được chụp trong điều kiện tốt, ánh sáng và nền không tối. Vì trình trích xuất vector đặc trưng sử dụng hệ số tương quan của khuôn mặt nên rất dễ bị nhạy cảm đối với những bức ảnh thật nhưng có độ sharpness không cao, hay những bức ảnh giả trên màn LCD có độ sharpness cao.



Hình 5: Ảnh trái là ảnh thật chụp trong điều kiện ánh sáng tối và bên phải là ảnh giả được lấy lại có độ sắc nét cao

3. Mô hình sẽ hoạt động tốt hơn nếu hai phân phối giữa hai class xa nhau, còn nếu hai phân phối giữa hai class gần nhau sẽ có một số phân loại nhầm không theo ý muốn. Ví dụ như khi gấp nhiều, blur .. thì sẽ nhận dạng sai do nhiều cũng được tính vào là yếu tố trên khuôn mặt.

- Để giải quyết triệt để vấn đề với mọi loại giả mạo, ta phải tìm một thuật toán tối ưu hơn, có độ chính xác đáng tin cậy trong mọi điều kiện môi trường sáng, tối. Không những thế phải có thể phân biệt một cách chính xác giữa hai phân phối live, spoof rất gần nhau.

Spoof Type						Illumination Condition and Environment						Normal		Strong		Back		Dark	
Print			Paper Cut			Replay			3D			Normal		Strong		Back		Dark	
Photo	Poster	A4	Face Mask	Upper Body Mask	Region Mask	PC	Pad	Phone	Mask	Indoor	Outdoor								

Hình 6: Các loại giả mạo khuôn mặt có thể có với điều kiện môi trường khác nhau

Vì bài toán này dựa trên cách tiếp cận Machine Learning nên một nhược điểm của nó là không thể chạy trên GPU, nên ta phải chuyển sang sử dụng các model neural network có thể inference trên GPU:

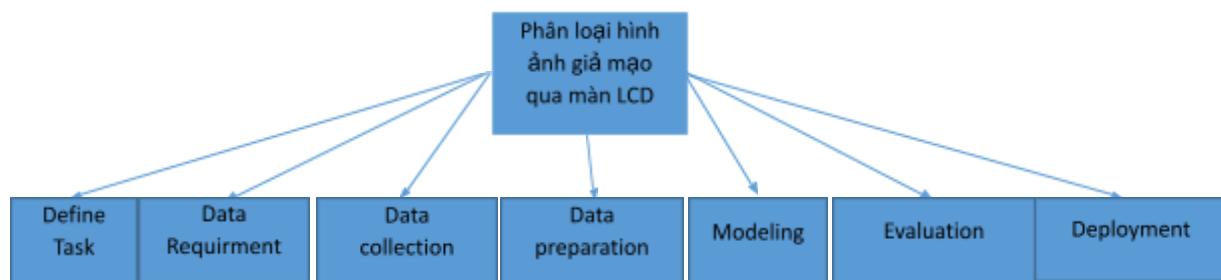
- Thêm các tính năng trích xuất đặc trưng của hình ảnh như 3D map, đặc trưng về cạnh, các đặc điểm về môi trường, điều kiện sáng.
- Sử dụng model CNN sâu trên hình ảnh 2D

Link:

<https://arxiv.org/pdf/2007.12342.pdf?fbclid=IwAR3UTXdUSNTtpWCOmiPbPbvAoVAlDLjzoveLgDwiB1yhd8SXtDwgJ7OfMsk>

II. Phương pháp

2.1 Tổng quan quá trình thực hiện



Hình 1 : Sơ đồ quá trình thực hiện dự án

Để thực hiện bất kì một bài toán áp dụng công nghệ AI nào, ta cần phải có một sơ đồ trình bày những mục tiêu cần thực hiện qua từng giai đoạn, với bài toán phát hiện khuôn mặt giả mạo này ta cần phải đưa ra những câu hỏi sau:

1. Define Task: Loại vấn đề ta đang cố gắng giải quyết ở đây là gì và tìm ra phương pháp tối ưu ?

Phát hiện hình ảnh được chụp lại qua LCD là một vấn đề kỹ thuật số quan trọng, vì việc chụp lại hình ảnh thường liên quan đến tạo ra một hình ảnh giả nhằm đánh lừa các hệ thống nhận dạng khuôn mặt, các hệ thống bảo mật,... Hiện nay có rất nhiều phương pháp phát hiện recaptured image như Eye Blink Check (1), phương pháp chống giả mạo bằng Challenge-Response (2), hay chống giả mạo bằng CNN classify kèm với 3D Map (3)....

Với phương pháp Eye Blink Check (1), sẽ kiểm tra xem khuôn mặt trong camera có nháy mắt hay không, nếu có nháy mắt -> real và ngược lại. Nhược điểm là sẽ cần

capture một số lượng lớn frame vì có người chớp nhiều, chớp ít nên không thể biết trước được.

Với phương pháp (2) hệ thống sẽ yêu cầu người dùng thực hiện các thao tác nào đó (Challenge) và người dùng phải thực hiện theo (response) để kiểm tra. Cụ thể máy có thể yêu cầu người dùng quay mặt sang phải, trái, lên trên, xuống dưới, ... Nhược điểm lớn nhất của cách làm này sẽ gây ra trải nghiệm không tốt cho người dùng do phải thực hiện khá nhiều thao tác trước khi được nhận diện.

Với phương pháp (3) có thể nói đây là phương pháp rất tốt để chống nhiều loại giả mạo khác nhau, sử dụng CNN model sẽ trích xuất các vùng ảnh nhỏ ra khỏi mặt và gán nhãn real/fake để train một mạng nhận diện riêng. Sau đó sẽ có một Auto Encoder Network riêng để suy ra 3D Map từ ảnh 2D đầu vào, mạng này được train sao cho với mặt thật thì nối khít còn với ảnh giả thì hoàn toàn phẳng. Từ đó sẽ dùng SVM để phân loại real/fake trên 3D Map cho ra kết quả cuối cùng. Tuy nhiên với phương pháp này độ khó của encoding rất cao mà hiện nay em vẫn đang nghiên cứu thêm để tự triển khai.

Vậy em đề xuất một phương pháp hiệu quả về mặt khái niệm để phát hiện hình ảnh được chụp lại, phương pháp được xây dựng dựa trên tính toán số liệu thống kê hình ảnh (4). Các đặc trưng là hệ số tương quan pixel tính qua các patch 5x5 của hình ảnh được transform qua miền vi phân. Sau đó được training trên Support Vector Machine (SVM) để ra kết quả cuối cùng. Phương pháp này giả định rằng phân phối giữa hai loại ảnh thật và giả không quá gần nhau.

2. Data Requirement:

- Ta cần những dữ liệu có cấu trúc, thành phần như thế nào?

Với yêu cầu như trên, thì bài toán có thể được coi như là phân loại nhị phân. Do đó dữ liệu của ta phải có hai class, class 0 (ảnh real) sẽ là ảnh live qua camera và class 1 (ảnh fake) sẽ là các loại giả mạo trên LCD như capture qua điện thoại, laptop.

3. Data collection: Chúng ta thu thập dữ liệu từ những phương pháp nào ?

Sau khi đã xác định được các thành phần cần có của dữ liệu tiến hành thu thập dữ liệu theo những cách sau đây.

- Method 1: Các tập dataset có label công khai
- Method 2: Tìm kiếm hình ảnh trên web và tải xuống theo cách thủ công
- Method 3: Data Augmentation (flipping, cropping, rotation)

Thực hiện 3 phương pháp trên ta có bộ dữ liệu như sau:

- Class 0: 150.000 image có độ phân giải từ thấp đến cao
- Class 1: 138.000 image được thu thập từ các loại giả mạo khác nhau trên màn hình điện thoại, laptop có độ phân giải từ thấp đến cao.

4. Data preparation: Chúng ta phải thực hiện những cách gì để xử lý dữ liệu đó ?

Sau khi có được dữ liệu cần thiết, có thể nói giai đoạn chuẩn bị dữ liệu cho model là công đoạn tốn nhiều công sức nhất, vì dữ liệu đến từ nhiều nguồn và từ nhiều cấu trúc khác nhau, không những thế dữ liệu có thể bị gắn nhãn sai gây ảnh hưởng không nhỏ

đến hiệu suất model trong quá trình training và testing. Vậy thực hiện xử lý những cách sau :

- Làm sạch dữ liệu hình ảnh, xử lý ảnh bị trùng nhau
- Crop lại toàn bộ khuôn mặt sử dụng lib Face-recognition.
- Loại bỏ các giá trị bị thiếu, loại bỏ các giá trị Nan feature vector của hình ảnh
- Loại bỏ các hình ảnh không cần thiết (hình ảnh quá mờ, chất lượng kém trong class 0 và 1) sử dụng hai phương pháp Laplacian và Fast Fourier Transform , tổ chức lại dữ liệu thành dữ liệu có cấu trúc để phù hợp với input của bài toán.

5. Modeling: Dữ liệu có thể được trích xuất theo cách nào để đi đến câu trả lời được yêu cầu?

- Giai đoạn này tập trung vào việc phát triển model và các thuật toán cùng metrics
- Thử nghiệm các thuật toán và model khác nhau sao cho đạt được độ chính xác tốt nhất.

Sau khi training(split 80%) và testing(split 20%) trên data có ở bước 3 thu được kết quả tốt nhất như sau:

- Testing set:

Patch size	Accuracy live	Accuracy spoof	Overall accuracy
5x5	90.5%	88.5%	90%

- Confuse Matrix trên testing:

0(True)	28113	2925
1(True)	3464	24480
0(Predict)	1(Predict)	

- Sau khi đánh giá trên tập testing tiến hành phân tích lỗi. Lấy 150 bức ảnh predict sai của class 1(True) và class 0(True) có thể phát hiện ngay ra vấn đề gây nhận diện sai ,2925 ảnh đều đến từ phân phối rất gần với class 1 tức là có rất nhiều hình ảnh mờ và rung và bao gồm rất nhiều hình ảnh bị gắn nhãn sai và đến từ những điều kiện môi trường ánh sáng quá tối. 3464 hình ảnh phần lớn do độ tương đồng với các bức ảnh real quá gần khiến cho model nhận diện nhầm thành real
- Thực hiện tuning lại model ta được kết quả:

- Testing set:

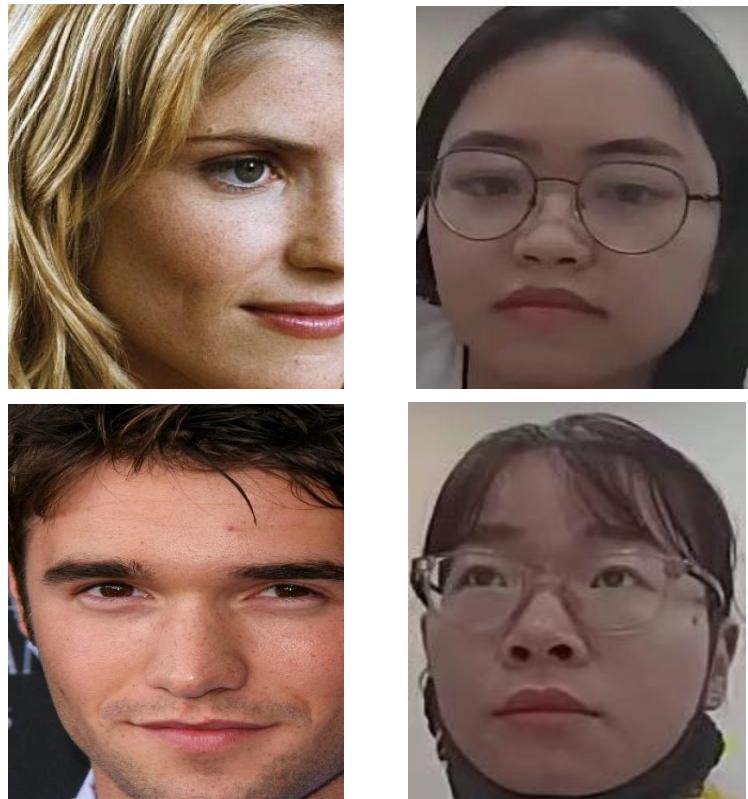
Patch size	Accuracy live	Accuracy spoof	Overall accuracy
5x5	94.5%	93.2%	94%

- Confuse Matrix trên testing:

0(True)	29688	1565
1(True)	2152	25628
0(Predict)	1(Predict)	

6. Evaluation: Model đã được training có thực sự giải quyết tốt bài toán với dữ liệu thực tế hay không ?

- Model evaluation được thực hiện trong suốt quá trình phát triển model và trước khi model đó được deployed
- Với model đã được training vấn đề ở đây là phải làm gì để đưa model đó hoạt động trong các dữ liệu thực tế trong hệ thống chấm công tại Công ty Edso Labs, tiến hành thu thập ảnh của nhân viên từ camera ip tại tầng 1 ta nhận thấy sự khác nhau giữa phân phối của các bức ảnh live qua camera ip với ảnh trong dataset thu thập từ bước 3:



Hình 2 : Sự khác nhau giữa ảnh thật giữa dữ liệu training và thực tế

- Sự khác biệt giữa những bức ảnh giả mạo được chụp từ camera ip không có gì khác nhau nhiều giữa hai phân phối dataset, một số ví dụ về ảnh fake được thu lại từ camera tầng 1 qua điện thoại:



Hình 3: ảnh trái là ảnh fake được thu qua camera và phải là ảnh fake được chụp qua điện thoại

- Tổng cộng thu thập 1200 hình ảnh live của nhân viên trong công ty, và 850 hình ảnh fake (ảnh chụp lại từ màn hình điện thoại, máy tính, và qua camera) của tất cả mọi người đó. Với tập dữ liệu thực tế này, ta sẽ fit lại model vừa train với tập dataset lớn trên để cho ra kết quả cuối cùng. Nhớ rằng định nghĩa class 0 là lớp quan trọng hơn class 1, vì ta không muốn hệ thống phân loại nhầm khuôn mặt live thành fake, và chấp nhận một số trường hợp phát hiện nhầm khuôn mặt fake thành live.
- Sau khi fit lại model trên dữ liệu thực tế tại công ty ta thu được kết quả sau:
 - Training set:

Patch size	Accuracy live	Accuracy spoof	Overall accuracy
5x5	99.7%	99%	99.4%

- Confuse matrix trên training set:

0(True)	749	2
1(True)	6	649
	0(Predict)	1(Predict)

- Testing set:

Patch size	Accuracy live	Accuracy spoof	Overall accuracy
5x5	97.3%	95.8%	96.5%

- Confuse matrix trên testing set:

0(True)	197	4
1(True)	8	143
	0(Predict)	1(Predict)

- Tiến hành thử nghiệm trên máy có CPU I7-10700, kết quả model chạy trung bình trên mỗi hình ảnh như sau:

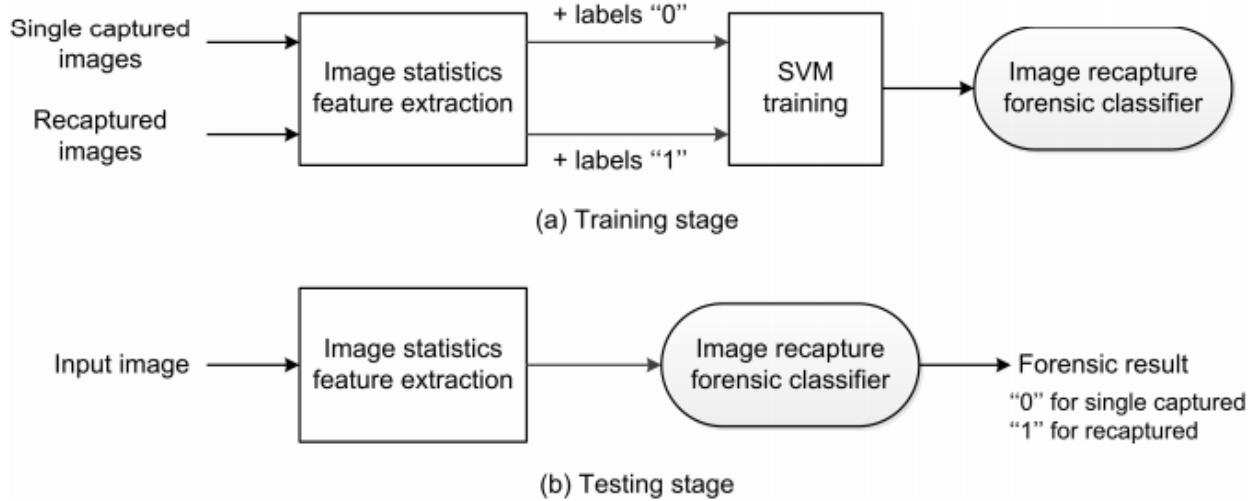
Feature extraction	SVM classification	Total
0.003s	0.001s	0.004s

7. Deployment: Một khi model đã được đánh giá và chúng ta đã tự tin về cách model hoạt động trong dữ liệu thực tế, nó sẽ được triển khai để đi vào hoạt động thực tế, phục vụ cho mọi người. Trong project này, sử dụng docker để đóng gói lại toàn bộ model và source code và gửi lên gitlab của Công ty.

Trong quá trình thực hiện bài toán, giai đoạn Data preparation và Modeling mất nhiều thời gian để thực hiện nhất, vì đối với bất cứ một bài toán nào sẽ có rất nhiều phương pháp và cách tiếp cận khác nhau để giải quyết và ta phải lựa chọn một phương pháp phù hợp, dễ triển khai và độ phức tạp của model không quá cao để tránh mất nhiều thời gian khi inference.

2.2 Kiến trúc model

Đối với phương pháp này, thực hiện theo hai stage như sau:



Hình 7: Sơ đồ tổng quan phương pháp

□ Stage 1:

- Đối với một hình ảnh nhất định, trích xuất một vector đặc trưng bao gồm các hệ số tương quan pixel trong miền vi phân hình ảnh. Vector này sau đó được đưa vào phục vụ cho phát hiện hình ảnh được recap.
- Các vector được trích xuất cùng với các label (0: no-recap và 1: recap) được đưa vào SVM để training

□ Stage 2: Trong quá trình testing với một hình ảnh nhất định, một vector đặc trưng được trích xuất và đưa vào model SVM đã được training để đưa ra dự đoán.

2.3 Image statistics feature extraction

Ý tưởng giả định rằng đặc trưng trích xuất của hình ảnh không phụ thuộc vào nội dung của hình ảnh mà chỉ nhạy cảm với sự khác biệt giữa hình ảnh live và spoof. Để đạt được mục tiêu này, một chiến lược phổ biến là loại bỏ các thành phần tần số thấp của ảnh và cho rằng thông tin phân biệt có trong thành phần tần số cao của bức ảnh.

Chính xác hơn, đối với một hình ảnh X nhất định có size MxN, trước hết ta áp dụng bộ lọc thông thấp, sau đó tính toán hai residual image, tức là sự khác biệt giữa X và ảnh sau khi áp dụng bộ lọc thông thấp của nó. Hai ảnh như:

$$R^{(i)} = \text{trim}(X - X * f^{(i)})$$

Trong đó * là tích chập toán học và chức năng trim lịa bỏ hàng và cột đầu tiên, cũng như hàng và cột cuối cùng khỏi hình ảnh đầu vào. Hai bộ lọc $f^{(1)}$ và $f^{(2)}$ được cho bởi:

$$f^{(1)} = [0, 0, 0; 1/2, 0, 1/2; 0, 0, 0] \text{ và } f^{(2)} = [0, 1/2, 0; 0, 0, 0; 0, 1/2, 0]$$

Hai hình ảnh residual image mô tả các cạnh và nhiễu trong hình ảnh, do đó từ residual image có thể trích xuất một vector đặc trưng thống kê có thể hiển thị những thay đổi trong một hình ảnh được thu lại.



Hình 8: 2 hình ảnh residual sau khi áp 2 bộ lọc

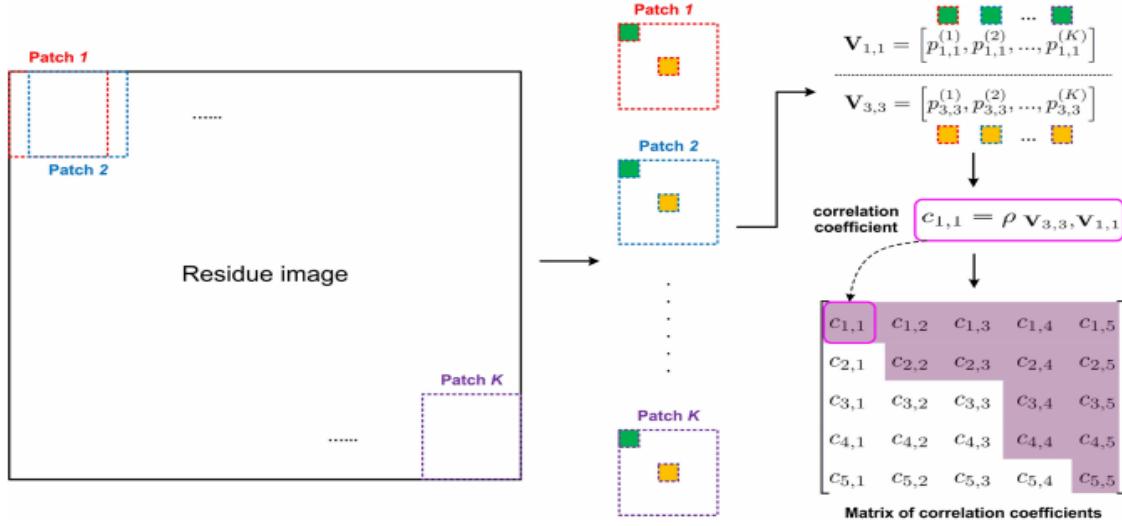
Có hai residual image ta cần tính toán các hệ số tương quan pixel giữa các residual. Từ một $R^{(i)}$, trước hết trích xuất tất cả các patch 5x5 chòng chéo của mỗi ảnh residual được biểu thị bằng $p^{(1)}, p^{(2)}, \dots, p^{(K)}$. Với K là tổng số các patch. Các phần tử mỗi $p^{(K)}$ được biểu thị bằng $P_{i,j}^K$ với $i,j \in \{1,2,3,4,5\}$. Sau đó chọn giá trị residual có cùng chỉ số i,j từ mỗi patch image và concat để tạo thành một vector V:

$$V_{i,j} = [P_{i,j}^1; P_{i,j}^2; \dots; P_{i,j}^K]$$

Tổng cộng ta có 25 vector như vậy. Sau đó tính toán hệ số tương quan $c_{i,j}$ giữa vector tương ứng ứng với pixel trung tâm tức là $V_{3,3}$ và tất cả 25 vector. Cụ thể :

$$c_{i,j} = \rho \mathbf{v}_{3,3} \cdot \mathbf{v}_{i,j} = \frac{\sum_{k=1}^K (p_{3,3}^{(k)} - \bar{p}_{3,3}) (p_{i,j}^{(k)} - \bar{p}_{i,j})}{\sqrt{\sum_{k=1}^K (p_{3,3}^{(k)} - \bar{p}_{3,3})^2} \sqrt{\sum_{k=1}^K (p_{i,j}^{(k)} - \bar{p}_{i,j})^2}},$$

Trong đó $\bar{p}_{3,3}$ và $\bar{p}_{i,j}$ lần lượt là giá trị TB của các phần tử trong $V_{3,3}$ và $V_{i,j}$. Với các giá trị c này sẽ mang lại khả năng phân biệt các thay đổi do recap image gây ra. Hình dưới đây minh họa quá trình trích xuất matrix các hệ số tương quan từ residual image :



Hình 9: Module trích xuất đặc trưng của hình ảnh dựa vào hệ số tương quan

Matrix hệ số tương quan được flatten thành một vector 24 chiều từ một trong hai ảnh residual. Do đó kích thước của vector đặc trưng cuối cùng là 48 chiều từ việc concat từ hai ảnh residual. Các vector này có được trên từ các hình ảnh đầu vào có nhãn label nhằm mục đích training bộ phân loại SVM.

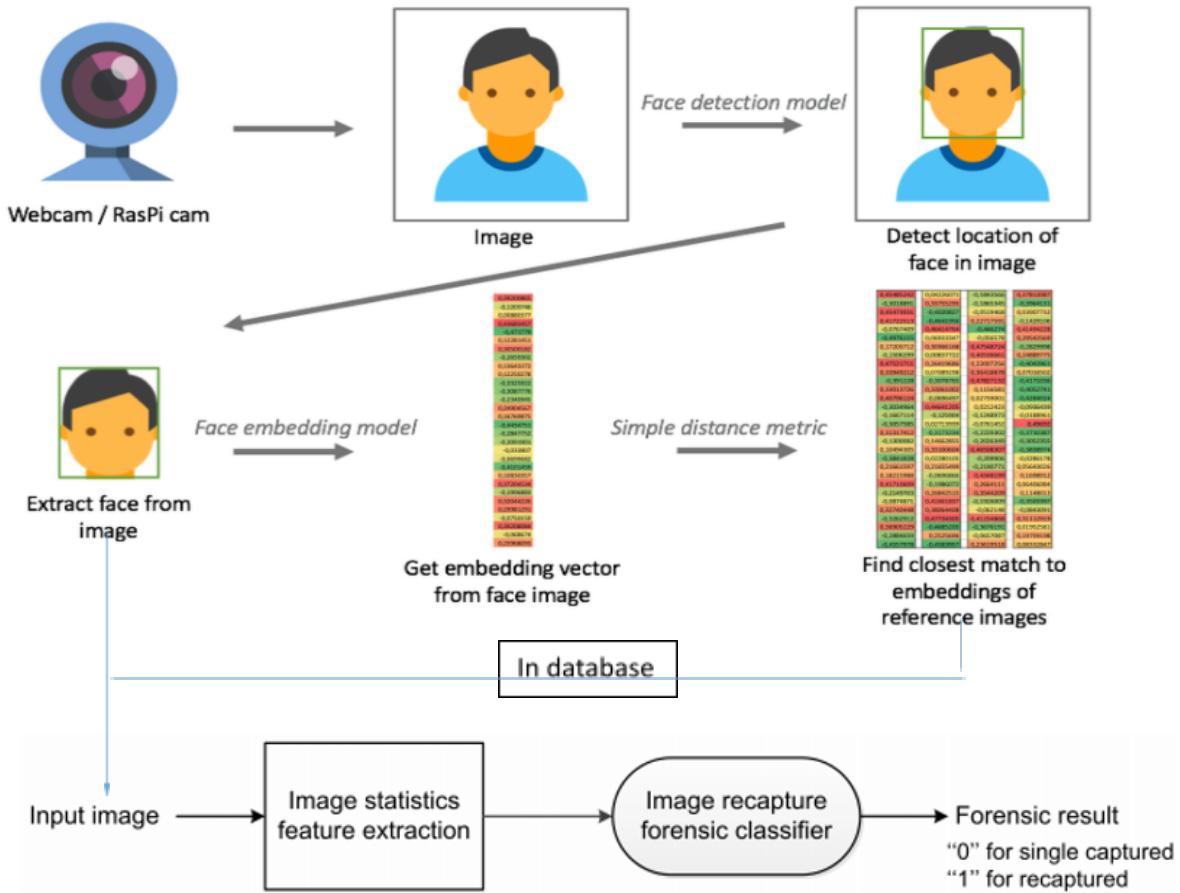
2.4 SVM classifier

Có một tập dữ liệu vector 48 chiều được đánh nhãn, sử dụng SVM để tiến hành training với các thông số sau:

- Training set và Testing set: 80-20
- Các vector trước khi đưa vào sẽ được chuẩn hóa về một khoảng nhất định với công thức: $z = (x-u)/s$
 - u: mean của training samples
 - s: standard deviation
- Gamma = 1
- Class_weight = {0:4,1:2}
- Kernel : RBF
- Metrics: accuracy, confuse matrix

Model về cơ bản nhận diện tốt đối với những hình ảnh live và spoof có phân phối xa nhau, đối với những hình ảnh spoof và live tương đồng nhau thì trình trích xuất đặc trưng sẽ xuất ra các vector 48 chiều với các giá trị khá gần nhau, điều này khiến cho quá trình classifier trên SVM gặp khó khăn trong việc phân biệt chính xác.

III. Đề xuất triển khai



Hình 10: Đề xuất cách tích hợp vào hệ thống nhận dạng khuôn mặt

Vì cách tiếp cận liên quan đến Machine Learning nên model không đòi hỏi chạy suy luận trên GPU. Với CPU dòng Intel Gen 5 trở lên có thể suy luận với tốc độ trung bình trên cả hai module extractor và classifier < 0.005s, các library và code có thể tương thích với Window và Linux.

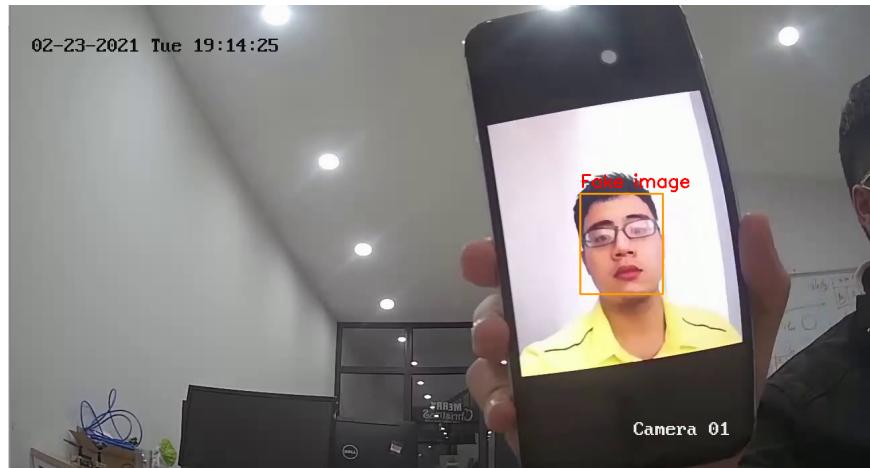
IV. Tổ chức dự án

Tôi chịu trách nhiệm phát triển chính, đưa ra các đề xuất về thuật toán, dataset và soạn các tài liệu về thuật toán và kết quả nghiên cứu.

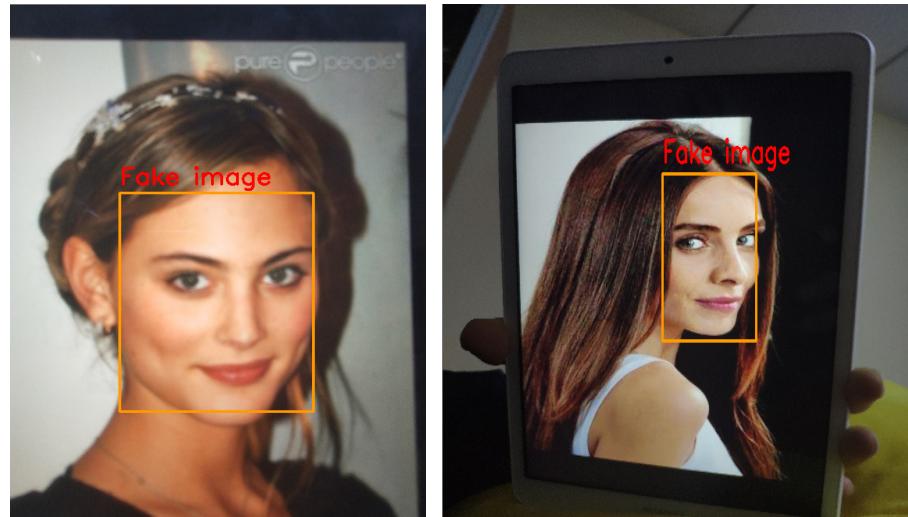
Anh Vũ Trung Kiên là team leader bộ phận RnD tại công ty sẽ chịu trách nhiệm quản lý tiến độ và hỗ trợ.

V. Ứng dụng và Kết luận

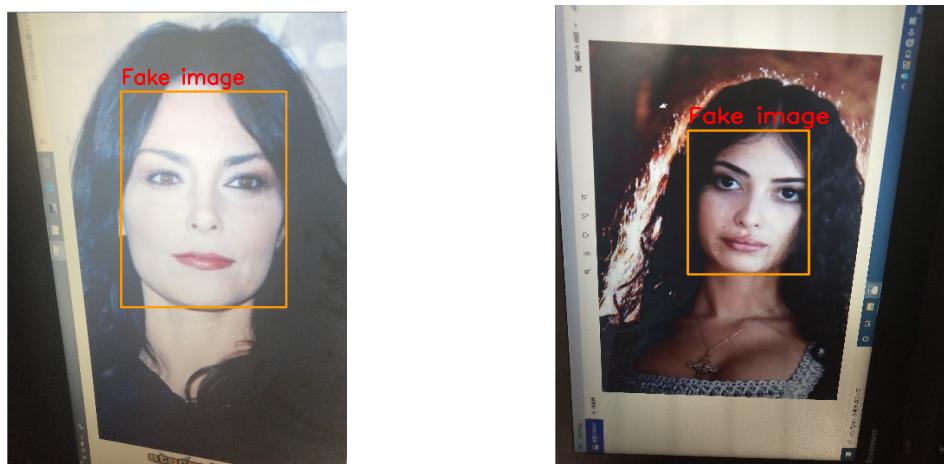
- Với kết quả thu được trong quá trình triển khai, ta có thể áp dụng model vào các loại phát hiện giả mạo qua màn hình điện thoại, laptop trong điều kiện ánh sáng tốt và hình ảnh thu được từ camera không quá mờ và rung.
 - Dữ liệu test ảnh fake được thử nghiệm :
 - + Hình ảnh chụp trên điện thoại, được đưa lên camera công ty cho ra một số kết quả như sau:



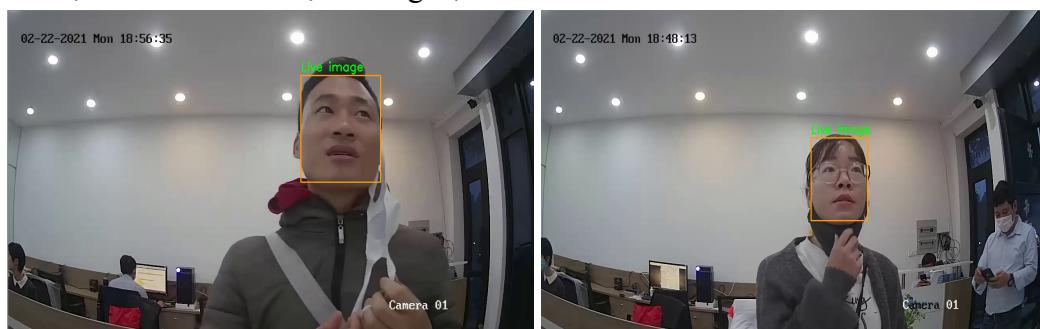
- + Hình ảnh chụp lại từ tablet:

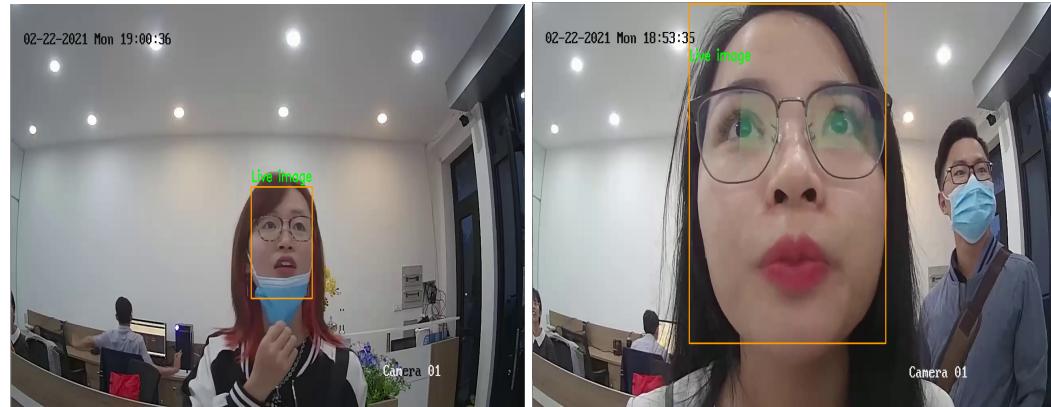


+ Hình ảnh chụp lại từ laptop



- Dữ liệu test ảnh live được thử nghiệm





- Hạn chế của bài toán này là chỉ chống được loại giả mạo cấp độ trên màn LCD, tôi sẽ tiếp tục nghiên cứu và phát triển để đề xuất một phương pháp tiếp cận mới dựa trên Deep learning phát hiện các loại giả mạo như trong hình 6, giúp đảm bảo an toàn cho các hệ thống an ninh về khuôn mặt.
- Trong báo cáo này, tôi đã trình bày một phương pháp thông kê dựa trên hình ảnh cho việc phát hiện hình ảnh chụp lại. Trình trich xuất đặc trưng hình ảnh rất nhanh và đơn giản, đạt được độ chính xác > 90% trên tập dữ liệu lớn về khuôn mặt CelebA và độ chính xác > 95% trên tập dữ liệu thực tế tại Công ty.

VI. Tài liệu tham khảo

1. http://www.gipsa-lab.fr/~kai.wang/papers/report_recap4n6.pdf
2. http://publications.idiap.ch/downloads/papers/2012/Chingovska_IEEEBIOSIG2012_2012.pdf?fbclid=IwAR09hDpBl8ZOw5HzyyymAT_e9r-lNh1DXxgRHdDOC5oRiU-kqjTZIDKvw5SA
3. <https://arxiv.org/pdf/1909.08848.pdf?fbclid=IwAR0wDtkhQ9GcOKC78se16qTx0q8Ei2zasirHI6tziSfeYrVjwrSq9hcNcgI>
4. https://arxiv.org/pdf/2007.12342.pdf?fbclid=IwAR0rPjKfVJvoaqQymTUb1ot_xPHinDNfisERqIPy9zQO8L6rDFENFtmwM
5. <https://www.pyimagesearch.com/2015/09/07/blur-detection-with-opencv/>
6. <https://www.pyimagesearch.com/2020/06/15/opencv-fast-fourier-transform-fft-for-blur-detection-in-images-and-video-streams/>