

Số:...../BKDT

Khoa: Điện – Điện tử

Bộ Môn: Viễn Thông

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

1. Họ và tên: Lê Hữu Hoàn MSSV: 1910190
2. Ngành: Điện - điện tử Chuyên ngành: Kỹ thuật Điện tử - Viễn thông
3. Đề tài: Nhận dạng video giả
4. Nhiệm vụ:

- Khảo sát các kỹ thuật tạo video giả
 - Phát triển thuật toán nhận dạng video giả
 - Phân tích và so sánh kết quả

5. Ngày giao nhiệm vụ: 21/06/2023
 6. Ngày hoàn thành nhiệm vụ: 28/08/2023

Họ và tên người hướng dẫn: Phần hướng dẫn
PGS.TS.Hà Hoàng Kha 100%
BM Viễn Thông, Khoa Điện - Điện Tử
Nội dung và yêu cầu Đồ án tốt nghiệp đã được thông qua Bộ Môn

TP.HCM, ngày 28 tháng 08 năm 2023

CHỦ NHIỆM BỘ MÔN

NGƯỜI HƯỚNG DẪN CHÍNH

PGS.TS.Hà Hoàng Kha

PGS.TS.Hà Hoàng Kha

**CÔNG TRÌNH ĐƯỢC HOÀN THÀNH TẠI
TRƯỜNG ĐẠI HỌC BÁCH KHOA –ĐHQG -HCM**

Cán bộ hướng dẫn Khóa luận tốt nghiệp : PGS.TS. Hà Hoàng Kha

Cán bộ chấm nhận xét 1 : (Ghi rõ họ, tên, học hàm, học vị và chữ ký)

Cán bộ chấm nhận xét 2 : (Ghi rõ họ, tên, học hàm, học vị và chữ ký)

Khóa luận tốt nghiệp được bảo vệ tại Trường Đại học Bách Khoa, ĐHQG Tp.HCM ngày 28 tháng 09 năm 2023

Thành phần Hội đồng đánh giá khóa luận tốt nghiệp gồm: (Ghi rõ họ, tên, học hàm, học vị của Hội đồng chấm bảo vệ khóa luận tốt nghiệp)

- 1.
- 2.
- 3.
- 4.
- 5.

Xác nhận của Chủ tịch Hội đồng đánh giá khóa luận tốt nghiệp và Chủ nhiệm Bộ môn sau khi luận văn đã được sửa chữa (nếu có).

CHỦ TỊCH HỘI ĐỒNG

CHỦ NHIỆM BỘ MÔN VIỄN THÔNG

LỜI CẢM ƠN

Trước hết, em xin cảm ơn thầy, cô bộ môn Viễn Thông, khoa Điện- Điện tử đã cho em cơ hội thực hành những lý thuyết đã học về Thị giác máy, Máy học, Xử lý ảnh. Đặc biệt, em xin cảm ơn đến PGS.TS. Hà Hoàng Kha đã đã nhiệt tình hỗ trợ em trong suốt quá trình nghiên cứu và thực hiện Đồ án tốt nghiệp.

Cũng nhân dịp này, em cũng xin cảm ơn đến thầy/ cô khoa Điện-điện tử, gia đình, bạn bè đã hỗ trợ, động viên em trong suốt quá trình thực hiện đồ án tốt nghiệp.

Em xin chân thành cảm ơn!

TP. HCM, ngày 28, tháng 09 năm 2023

Lê Hữu Hoàn

LỜI CAM ĐOAN

Tôi tên: Lê Hữu Hoàn, là sinh viên chuyên ngành Kỹ thuật Điện tử - Viễn thông, khóa 2019, tại Đại học Quốc gia thành phố Hồ Chí Minh – Trường Đại học Bách Khoa. Tôi xin cam đoan những nội dung sau đây là sự thật: (i) Công trình nghiên cứu này hoàn toàn do chính tôi thực hiện; (ii) Các tài liệu và trích dẫn trong luận văn này được tham khảo từ các nguồn thực tế, có uy tín và độ chính xác cao; (iii) Các số liệu và kết quả của công trình này được tôi tự thực hiện một cách độc lập và trung thực.

TP. HCM, ngày 28, tháng 09 năm 2023

Lê Hữu Hoàn

TÓM TẮT ĐỒ ÁN TỐT NGHIỆP

Công nghệ Deepfake cho phép tạo ra các video không có thật dựa trên việc sử dụng khuôn mặt. Với sự phát triển của các công cụ Deepfake, chúng đã được ứng dụng rộng rãi trong lĩnh vực giải trí và truyền thông. Có nhiều công cụ dựa trên mạng tạo sinh đối nghịch (GAN) như faceswap, puppet-mastery,...được sử dụng để tạo ra các video Deepfake. Tuy nhiên, một số cá nhân đã lợi dụng công nghệ Deepfake cho mục đích xấu, sử dụng để lừa đảo hoặc hạ bệ người khác. Do đó, việc phát triển các thuật toán để phát hiện video giả có vai trò quan trọng trong việc ngăn chặn những mục đích xấu và có thể được áp dụng trong lĩnh vực định danh cá nhân (eKYC). Sử dụng các tín hiệu sinh học như nhịp tim, nhịp thở,... đã được chứng minh có hiệu quả trong việc nhận biết video giả. Vì vậy, phương pháp này được sử dụng trong đồ án như một đặc trưng cho mô hình học máy.

Đồ án tốt nghiệp này sử dụng cấu trúc mạng Thin-Plate Spline Motion, chỉ cần một video và một hình ảnh để tạo ra các video đã được chỉnh sửa để phục vụ mục đích khảo sát nhận dạng video thật/ giả. Các tín hiệu Remote Photoplethysmography (rPPG) và thực hiện biến đổi Fourier tín hiệu này trong miền tần số để tạo ra các đặc trưng đầu vào cho mô hình học sâu. Các phương pháp hiện tại của rPPG chủ yếu tập trung vào mạng Mạng nơ-ron tích chập (CNN) và chưa có nhiều nghiên cứu về mạng Mạng nơ-ron hồi quy (RNN). Do đó, đồ án tốt nghiệp cũng tập trung vào nghiên cứu RNN và phát triển mô hình học sâu dựa trên cấu trúc mạng này. Kết quả đạt được là độ chính xác 82,23% trên tập dữ liệu kiểm thử. Sự kết hợp giữa các đặc trưng sinh học và chuyển động khuôn mặt bên ngoài của video cũng được xem xét. Đồ án tốt nghiệp này cũng đã cải tiến mô hình trên bằng cách kết hợp với mạng CNN dựa trên bề ngoài của khuôn mặt, và kết quả đạt độ chính xác 87,61% trên tập dữ liệu kiểm thử.

ABSTRACT

Deepfake technology enables the creation of non-authentic videos using the faces of different individuals. With the advancement of Deepfake tools, they have found extensive application in the realms of entertainment and media. Numerous tools based on Generative Adversarial Networks (GANs) like faceswap, puppet-mastery, etc are used to create fake video. have been developed to generate realistic-looking fabricated videos. However, some individuals have misused Deepfake technology for malicious purposes, including deception and defamation of others. Consequently, the development of algorithms to detect fake videos plays a crucial role in preventing such malicious intentions and can be applicable in the field of personal identification (eKYC). Utilizing biometric signals such as heart rate, respiration, etc. has been proven effective in recognizing fake videos. Hence, this method is employed in the project as a feature for machine learning models.

This project utilizes the Thin-Plate Spline Motion network architecture, which requires only a video and an image to create manipulated videos intended for the purpose of authentic/fake video identification. Remote Photoplethysmography (rPPG) signals are employed, and Fourier transforms are applied in the frequency domain to generate input features for the deep learning model. Current rPPG methods mainly focus on Convolutional Neural Networks (CNNs), and there is relatively less research on Recurrent Neural Networks (RNNs) in this context. Therefore, the project also delves into the study of RNNs and develops a deep learning model based on this network architecture. The achieved result is an accuracy of 82.23% on the test dataset. The combination of biometric features with facial appearance is also examined. The project further enhances the model by incorporating a CNN for facial appearance prediction, resulting in an accuracy of 87.61% on the test dataset.

Mục lục

1 GIỚI THIỆU	1
1.1 Đặt vấn đề	1
1.2 Phạm vi và phương pháp nghiên cứu	2
1.2.1 Phạm vi	2
1.2.2 Phương pháp nghiên cứu	2
1.3 Các đóng góp của đồ án	2
1.4 Bố cục đồ án tốt nghiệp	3
2 TỔNG QUAN VỀ DEEPFAKE VÀ CÁC NGHIÊN CỨU NHẬN DẠNG VIDEO GIẢ	4
2.1 Tổng quan các nghiên cứu Deepfake	4
2.2 Mô hình Thin-Plate Spline Motion Model for Image Animation	6
2.3 Tổng quan các nghiên cứu về nhận dạng video giả	7
2.4 Kết luận chương	8
3 BIẾN ĐỔI FOURIER VÀ BỘ LỌC	9
3.1 Biến đổi Fourier rời rạc	9
3.2 Mật độ phô	11
3.3 Bộ lọc tín hiệu	11
3.4 Kết luận chương	13
4 CƠ SỞ SINH HỌC VÀ KỸ THUẬT TRÍCH XUẤT REMOTE PHOTO-PLETHYSMOGRAPHY (rPPG)	15
4.1 Cơ sở sinh học	15
4.2 Thuật toán	17
4.2.1 Green channel based rPPG	17
4.2.2 Chrominance based rPPG	18
4.2.3 Plane-Orthogonal-to-Skin (POS)	21
4.2.4 Local group invariance (LGI)	23
4.3 So sánh giữa các thuật toán	25
4.4 Lựa chọn vùng ROI cho rPPG	26
4.5 Kết luận chương	27

5 MỘT SỐ MÔ HÌNH MÁY HỌC VÀ CÁC PHƯƠNG PHÁP TỐI ƯU & ĐÁNH GIÁ MÔ HÌNH	28
5.1 Một số mô hình máy học	28
5.1.1 Support Vector Machine	28
5.1.2 Artificial Neural Network	30
5.1.3 Convolutional Neural Network	32
5.1.4 Recurrent Neural Network	33
5.2 Thuật toán tối ưu	36
5.2.1 Gradient Descent	36
5.2.2 Gradient Descent with Momentum	37
5.2.3 RMS prop	38
5.2.4 Adam	38
5.3 Deep supervision	39
5.3.1 Hidden Layer Deep Supervision (HLDS)	39
5.3.2 Different Branches Deep Supervision (DBDS)	40
5.3.3 Deep Supervision Post Encoding	40
5.4 Thông số đánh giá	41
5.4.1 Confusion matrix	41
5.4.2 Precision	42
5.4.3 Recall	42
5.4.4 F1-Score	43
5.4.5 Area Under the Curve (AUC)	43
5.5 Kết luận chương	44
6 MÔ HÌNH LSTM SỬ DỤNG TÍN HIỆU RPPG	45
6.1 Dữ liệu	45
6.2 Thuật toán đề xuất	46
6.2.1 Lựa chọn thuật toán trích xuất rPPG	46
6.2.2 Thiết kế mô hình	46
6.3 Kiểm thử thuật toán và cải tiến	53
6.4 Kết luận chương	53
7 KẾT QUẢ VÀ PHÂN TÍCH	54
7.1 Phương pháp tiếp cận	54
7.2 Thiết lập thông số	54
7.3 Kết quả và phân tích	55
7.3.1 Kích thước phân đoạn ω	55
7.3.2 Khảo sát phương pháp trích xuất rPPG	55
7.3.3 Khảo sát đặc trưng đầu vào	56

7.3.4	Mô hình LSI+ LSTM	57
7.3.5	Mô hình LSTM kết hợp với CNN	58
7.3.6	So sánh mô hình	60
7.3.7	So sánh với các dạng video Deepfake	61
7.4	Kết luận chương	62
8	KẾT LUẬN	63
8.1	Tóm tắt và kết luận chung	63
8.1.1	Những đóng góp của đề tài	63
8.1.2	Những hạn chế	64
8.2	Hướng phát triển	64
A	CODE THỰC HIỆN	69
A.1	Code trích xuất rPPG:	69
A.2	Mô hình huấn luyện:	70

Danh sách bảng

3.1	Biểu thức tương ứng với các bậc	13
4.1	So sánh giữa các phương pháp trích xuất rPPG	26
4.2	Các vùng ROI ứng với các phương pháp khác nhau	26
6.1	Cấu trúc dữ liệu	45
7.1	Khảo sát chiều dài phân đoạn	55
7.2	So sánh giữa các phương pháp trích xuất rPPG	55
7.3	So sánh các loại đặc trưng	56
7.4	Kết quả quá trình huấn luyện	57
7.5	Kết quả quá trình huấn luyện mô hình LSTM+CNN	59
7.6	So sánh mô hình với các phương pháp khác	61
7.7	So sánh hiệu quả với các phương pháp khác	61

Danh sách hình vẽ

2.1	Mạng GAN tạo Face swap	5
2.2	Mô hình Thin-Plate Spline Motion	6
3.1	Tín hiệu trong miền thời gian và miền tần số	10
4.1	Body Plethysmography	16
4.2	Sự hấp thụ và phản xạ của tế bào	16
4.3	Khảo sát mức độ hấp thụ và phản xạ ánh sáng	17
4.4	Mô hình phản xạ	18
4.5	Minh họa phép chiếu mặt phẳng	24
4.6	Khảo sát các vùng ROI	27
5.1	Minh họa cho thuật toán SVM	29
5.2	Mạng ANN đơn giản	30
5.3	Mô hình Multi-Layer Neural Network	31
5.4	Quá trình lan truyền thuận & nghịch	32
5.5	Quá trình tính tích chập trong CNN	33
5.6	Cấu trúc mạng RNN	34
5.7	Mô hình LSTM	35
5.8	Mô hình hồi quy tuyến tính	36
5.9	Thuật toán Gradient Descent	37
5.10	Hạn chế của Gradient Descent	37
5.11	Hidden Layer Deep Supervision	39
5.12	Different Branches Deep Supervision	40
5.13	Deep Supervision Post Encoding	41
5.14	Ma trận nhầm lẫn	42
5.15	AUC và đường cong ROC	43
6.1	Mô hình sử dụng LSTM	46
6.2	Vùng ROI	47
6.3	Mô hình chi tiết	48
6.4	Mô hình cải tiến kết hợp với đặc trưng bề ngoài	49
6.5	Mô hình chi tiết kết hợp LSTM và CNN đoạn đầu	50

6.6	Mô hình chi tiết kết hợp LSTM và CNN đoạn giữa	51
6.7	Mô hình chi tiết kết hợp LSTM và CNN đoạn cuối	52
7.1	Quá trình huấn luyện mô hình	57
7.2	Ma trận nhầm lẫn với tập test	58
7.3	Quá trình huấn luyện mô hình LSTM+CNN	59
7.4	Ma trận nhầm lẫn với tập test mô hình cải tiến	60
7.5	So sánh giữa các mô hình tạo video Deepfake	62
8.1	Video Presentation Attack Detection	65

DANH MỤC TỪ VIẾT TẮT

Ký hiệu	Tiếng Anh	Ý nghĩa tiếng Việt
BSS	Blind Source Separation	Tách nguồn mù
CHROM	Chrominance	Sắc thái màu
CNN	Convolutional neural network	Mạng nơ-ron tích chập
LGI	Local group invariance	Nhóm lân cận bất biến
LSTM	Long short time memory	Bộ nhớ dài - ngắn hạn
POS	Plane Orthogonal-to-Skin	Mặt phẳng trực giao với da
PSD	Power spectral density	Mật độ phổ công suất
rPPG	Remote Photoplethysmography	Tín hiệu Plethysmography từ xa sử dụng hình ảnh
RNN	Recurrent neural network	Mạng nơ-ron hồi quy
SVM	Support vector machine	Máy vector hỗ trợ

Chương 1

GIỚI THIỆU

Trong chương này sẽ trình bày về lý do chọn đề tài, nhiệm vụ đề tài và các phương pháp thiết kế nghiên cứu đồ án. Các chương tiếp theo sẽ lần lượt giải quyết các vấn đề đặt ra ở Chương 1 thông qua việc tìm hiểu cơ sở lý thuyết và tiến hành thực hiện các mô hình nhận dạng video giả.

1.1 Đặt vấn đề

Deepfake được ra đời với mục đích ban đầu áp dụng trong giải trí truyền thông. Tuy nhiên, một số bộ phận đã sử dụng Deepfake trong các mục đích xấu như tạo dữ liệu để lừa đảo, ghép ảnh người vào các hình ảnh nhạy cảm,...Do đó, gần đây rất nhiều nghiên cứu ra đời để nhận dạng ảnh do Deepfake tạo ra. Các phương pháp tập trung vào việc phát hiện các chi tiết lỗi trên khuôn mặt, ví dụ: lỗi trên vị trí môi, mắt, lỗi tái tạo tóc, râu,...Hoặc các phương pháp tập trung vào nhận dạng các điểm khác thường trong các hoạt động, ví dụ: tần suất chớp mắt, chuyển động của miệng,...Các phương pháp đó đạt được hiệu quả nhất định, tuy nhiên, các video Deepfake dần được cải tiến và ngày càng ít lỗi trong việc tạo video. Do đó, cần có những phương pháp khác hiệu quả hơn. Các tín hiệu sinh học được xem là một phương pháp mới và có hiệu quả cao trong việc xác thực video thật/ giả. Intel đã ứng dụng “blood blow” sử dụng cơ chế sinh học trong việc nhận dạng với độ chính xác 96%. Điều này chứng minh rằng sử dụng các sinh trắc sinh học có thể được ứng dụng trong lĩnh vực này. Phương pháp sử dụng phổ biến dựa trên tín hiệu Photoplethysmography. Do đó, đồ án tốt nghiệp này sử dụng tín hiệu này để thực hiện việc phân loại video thật/ giả.

Ngoài ra, các phương pháp phát hiện Deepfake dựa trên tín hiệu sinh học rPPG dựa trên mô hình máy học có hai phương pháp chính: sử dụng mạng CNN và RNN. Trong đó, các nghiên cứu về CNN nổi bật hơn nhiều, ít có những nghiên cứu về RNN. Do đó đồ án cũng thực hiện việc khảo sát việc thực hiện trên RNN. Đồng thời so sánh hiệu quả với các mô hình khác sử dụng CNN và SVM. Câu hỏi nghiên cứu đặt ra của đồ án là:

- Việc sử dụng tín hiệu sinh học rPPG có hiệu quả trong việc nhận dạng video thật/ giả

hay không?

- Phương pháp RNN có đạt hiệu quả như phương pháp sử dụng CNN và SVM như các công trình trước đó.

1.2 Phạm vi và phương pháp nghiên cứu

1.2.1 Phạm vi

Khảo sát mức độ chính xác dựa trên tập dữ liệu video giả được tạo bởi mạng Thin-Plate Spline Motion Model for Image Animation. Các video gốc được tác giả lấy mẫu và thu thập trên tập dữ liệu Faceforensics++ [16] và dữ liệu Celeb DF [22]. Lý giải cho việc sử dụng Thin-Plate Spline Motion để khảo sát do mạng này không yêu cầu dữ liệu đầu vào quá phức tạp: chỉ cần 1 video làm mẫu và 1 ảnh. Từ đó, có thể tạo ra các đoạn video giả, điều này, rất gần với thực tế, các nhóm tội phạm thông thường chỉ có được những ảnh đơn lẻ của một người để giả dạng.

1.2.2 Phương pháp nghiên cứu

Các thuật toán được thực hiện dựa trên những tài liệu, nghiên cứu trước đó. Từ đó, dựa trên các phương pháp toán học: dựa trên các lý thuyết về xử lý tín hiệu, máy học, học sâu để xây dựng và phát triển thuật toán. Đánh giá độ hiệu quả và chính xác dựa trên các phương pháp thống kê, thực tiễn.

1.3 Các đóng góp của đồ án

Đồ án thực hiện một số đóng góp sau so với các công trình trước đó.

- Khảo sát các mô hình tạo video giả (Deepfake video)
- Xây dựng mô hình và cải tiến độ chính xác của mô hình Long Short Term Memory kết hợp với cơ chế Deep Supervision cho tín hiệu sinh học Remote Photoplethysmography.
- So sánh với các mô hình khác.
- So sánh hiệu quả giữa các phương pháp trích xuất rPPG trong nhận dạng video thật/giả.

1.4 Bố cục đồ án tốt nghiệp

Đồ án tốt nghiệp được chia thành 8 chương với các nội dung như bên dưới:

- **Chương 1: GIỚI THIỆU**

Chương này sẽ trình bày về tính cấp thiết của đề tài, phạm vi nghiên cứu, và một số đóng góp của đề tài

- **Chương 2: DEEPFAKE VÀ TỔNG QUAN CÁC NGHIÊN CỨU VỀ NHẬN DẠNG VIDEO GIẢ**

Chương này sẽ tổng quan tình hình nghiên cứu về Deepfake và nhận dạng video giả được tạo từ Deepfake. Ngoài ra, chương này cũng đề cập đến mô hình được sử dụng để tạo video giả cho bộ dữ liệu huấn luyện mô hình.

- **Chương 3: BIẾN ĐỔI FOURIER VÀ BỘ LỌC TÍN HIỆU**

Chương này trình bày biến đổi Fourier, năng lượng của tín hiệu và các bộ lọc tín hiệu. Chương này là cơ sở để đánh giá tính hiệu trong miền tần số cho Chương 4 và được sử dụng như là một đặc trưng đầu vào cho mô hình ở Chương 6.

- **Chương 4: CƠ SỞ SINH HỌC VÀ KỸ THUẬT TRÍCH XUẤT REMOTE PHOTOPLETHYSMOGRAPHY (rPPG)**

Chương này trình bày về cơ sở lý thuyết của Remote Photoplethysmography và cơ sở lý luận cho việc sử dụng rPPG cho việc phân biệt video thật/ giả. Ngoài ra, chương này cũng chú trọng phân tích các kỹ thuật để trích xuất rPPG.

- **Chương 5: MỘT SỐ MÔ HÌNH MÁY HỌC VÀ CÁC PHƯƠNG PHÁP TỐI ƯU & ĐÁNH GIÁ MÔ HÌNH**

Chương này trình bày lý thuyết của một số mô hình máy học phổ biến và các phương pháp cải thiện độ chính xác, tốc độ của mô hình.

- **Chương 6: MÔ HÌNH LSTM SỬ DỤNG TÍN HIỆU RPPG**

Chương này thực hiện việc xây dựng mô hình trên cơ sở lý thuyết từ Chương 2 đến Chương 5.

- **Chương 7: KẾT QUẢ VÀ PHÂN TÍCH**

Chương này thực hiện việc đánh giá mô hình ở Chương 6, và thực hiện các so sánh để tìm được thông số của mô hình. Để đảm bảo tính khách quan, mô hình ở Chương 6 cũng được so sánh với các mô hình đã được nghiên cứu trước đó.

- **Chương 8: KẾT LUẬN**

Chương này sẽ tổng kết toàn bộ đề tài và đưa ra nhận xét về ưu và khuyết điểm của đề tài. Từ đó đồ án cũng đề xuất các hướng phát triển mới cho đề tài.

Chương 2

TỔNG QUAN VỀ DEEPFAKE VÀ CÁC NGHIÊN CỨU NHẬN DẠNG VIDEO GIẢ

Từ Chương 2 đến Chương 5 sẽ trình bày về cơ sở các lý thuyết được ứng dụng trong mô hình phân loại video thật/ giả. . Chương này sẽ trình bày về tổng quan về các nghiên cứu về nhận dạng video giả và tìm hiểu về khái niệm Deepfake là gì? Từ cơ sở này, đồ án sẽ xây dựng bộ dữ liệu video thật và giả để thử nghiệm với các mô hình máy học.

2.1 Tổng quan các nghiên cứu Deepfake

Deepfake là một dạng video, hình ảnh hoặc âm thanh giả. Đa phần các Deepfake được tạo ra bởi mạng tạo sinh đối nghịch (Generative Adversarial Network). Về nguyên tắc hoạt động, dữ liệu đầu vào gồm 2 loại dữ liệu: mục tiêu (dữ liệu khuôn mặt muốn ghép vào) và nguồn (ảnh/video mẫu). Mạng GAN gồm 2 mạng con nhỏ hơn:

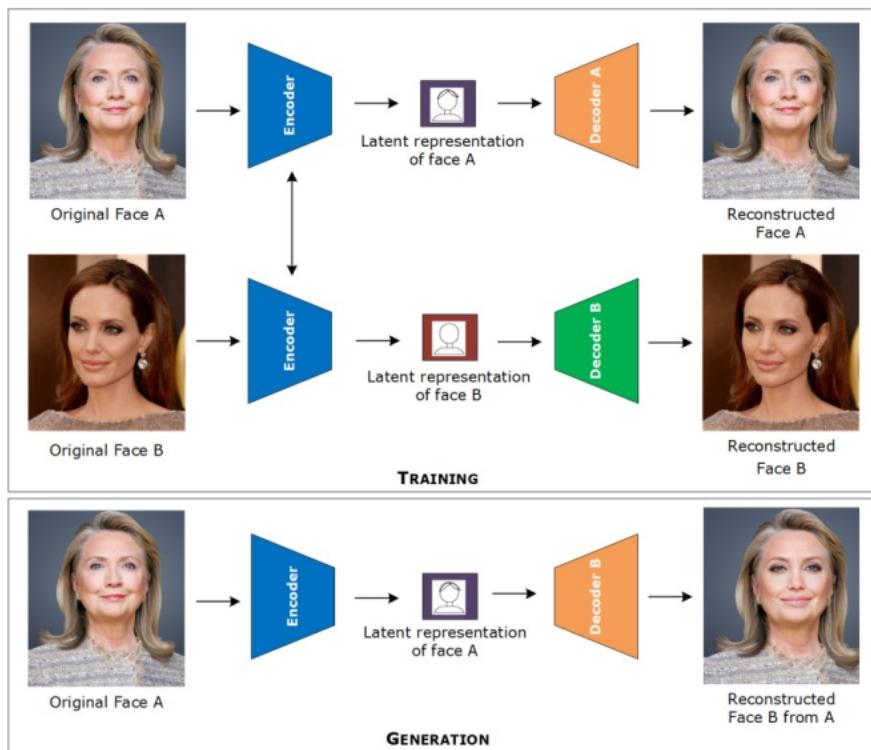
- **Mô hình sinh:** Dùng để sinh ra ảnh/ video Deepfake
- **Mô hình phân biệt:** Dùng để xác định xem ảnh/video Deepfake có giống với ảnh/video thật hay không.

Một số dạng Deepfake phổ biến được tạo bởi mạng GAN [13]:

- **Face Swap:** dạng này được dùng để hoán đổi khuôn mặt giữa 2 người. Nói cách khác, lấy khuôn mặt của người A và gắn vào video của người B. Hình 2.1 minh họa cho cách hoạt động của Face Swap.
- **Puppet-Mastery:** dạng này tập trung tạo ra các chuyển động, biểu lộ cảm xúc của khuôn mặt

- **Lip-syncing:** dạng này tạo ra các chuyển động của miệng sao cho phù hợp với âm thanh được đưa vào.
- **Entire Face Synthesis:** Mạng này tạo ra khuôn mặt người từ việc tổng hợp các chi tiết khuôn mặt, bao gồm mắt, mũi, miệng, và các đặc điểm khác của khuôn mặt để tạo thành một hình ảnh hoàn chỉnh và tự nhiên.
- **Facial Attribute Manipulation:** Dựa vào video/ hình ảnh đầu vào, mạng này thực hiện việc sửa đổi các thuộc tính của khuôn mặt: màu da, tóc, kích thước mắt, cằm,... Đây là dạng thường gặp trong các ứng dụng chỉnh sửa như làm sáng khuôn mặt, mịn da, các bộ lọc

Hình bên dưới minh họa cách mạng GAN hoạt động với Face swap:



Hình 2.1: Mạng GAN tạo Face swap

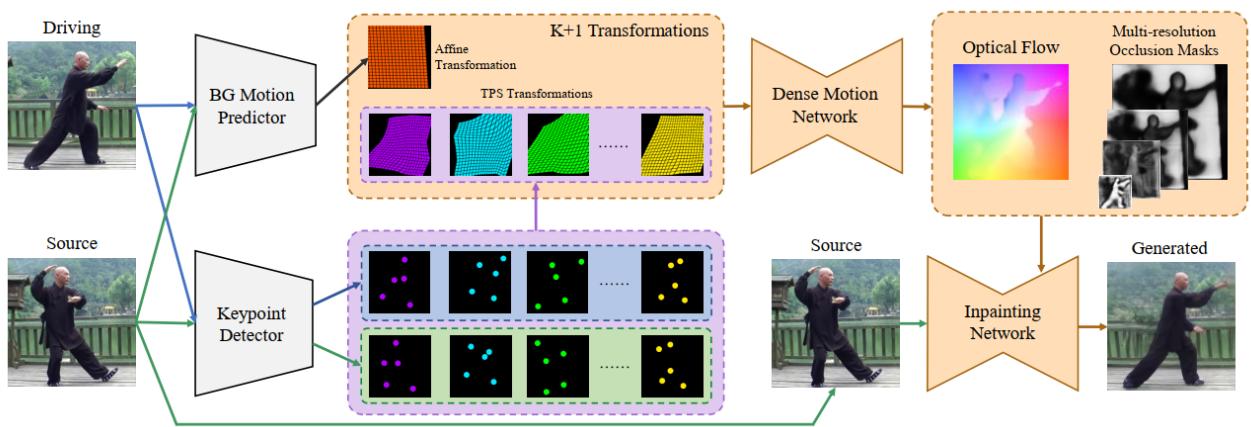
Hình 2.1 mô tả quá trình Face Swap gồm 2 quá trình chính: đầu tiên, sẽ rút trích đặc trưng khuôn mặt từ 2 đối tượng (video/ ảnh). Sau đó, thực hiện việc tạo ra các ảnh/ video face swap dựa trên các đặc trưng đó.

Mục đích ban đầu của Deepfake được ứng dụng trong giải trí, tuy nhiên, gầy đây nó đã được sử dụng không đúng mục đích nhằm lừa đảo, tạo tin giả,...

2.2 Mô hình Thin-Plate Spline Motion Model for Image Animation

Mạng Thin-Plate Spline Motion Model for Image Animation sẽ được sử dụng để tạo dữ liệu video giả trong việc khảo sát thuật toán nhận dạng video giả. Do đó, trong khuôn khổ đề án này, mô hình này sẽ được giới thiệu một cách khái quát mà không giới thiệu chi tiết các biểu thức tính toán hay cách xây dựng mô hình. Mạng Thin-Plate Spline Motion Model for Image Animation được sử dụng với dữ liệu đầu vào gồm 1 ảnh và 1 video mẫu (driving video). Thuật toán sẽ xử lý và học các chuyển động quang học (optical flow) trên video mẫu. Sau đó, sử dụng một mạng tạo sinh (GAN) áp dụng lên ảnh video đầu vào.

Theo [23], cấu trúc mạng được mô tả như hình sau:



Hình 2.2: Mô hình Thin-Plate Spline Motion

Như Hình 2.2, cấu trúc mạng bao gồm:

- **Keypoint Detector:** bộ phát hiện này dùng để xác định các điểm đặc trưng quan trọng trong dữ liệu đầu vào (Source và Driving). Các keypoint được sử dụng cho các biến đổi Thin-Plate Spline. Biến đổi Thin-Plate Spline được dùng để biến đổi các điểm keypoint từ không gian nguồn (Source) sang không gian đích (Driving).
- **BG Motion Predictor:** Thông thường nền (background) chiếm một phần lớn hơn so với đối tượng (foreground), giữa các frames của video có thể có sự chuyển động của nền. Điều này, có thể làm ảnh hưởng đến quá trình ước tính chuyển động của đối tượng. Do đó, mạng này được sử dụng để ước tính các biến đổi affine của nền (background): tức chuyển động của nền. [17]
- **Dense Motion Network:** mạng này được dùng để ước lượng chuyển động quang học (Optical flow) và thực hiện việc Multi-resolution Occlusion Masks. Trong đó, các chuyển

động quang học đặc trưng cho các chuyển động của video. Các Multi-resolution Occlusion Masks được dùng để che phủ một số đặc trưng trên feature map với các tỉ lệ khác nhau để tăng cường khả năng học của thuật toán.

- **Impainting Network:** Mạng này kết hợp với ngõ ra của Dense Motion Network và ngõ vào video gốc (Source) để tạo ra ngõ ra của mô hình.

Như vậy mạng này được dùng để tạo video giả từ hình ảnh. Mô hình này được cho là dễ sử dụng và phù hợp với tình huống thực tế trong các tình huống lừa đảo, mạo danh người khác.

Tóm lại, phần trên đã giới thiệu qua các công nghệ Deepfake gần đây và giới thiệu về mô hình tạo video giả được sử dụng trong đồ án này. Tiếp theo, đồ án sẽ giới thiệu về tổng quan các nghiên cứu về nhận dạng video giả.

2.3 Tổng quan các nghiên cứu về nhận dạng video giả

Các công nghệ phát hiện video giả có nhiều mô hình nghiên cứu khác nhau. Thông thường, các phương pháp tập trung nhận diện video thật/ giả dựa trên các đặc trưng trên khuôn mặt như: mắt, mũi, miệng,...hoặc dựa trên các chuyển động, tần suất của cơ mặt, chớp mắt, chuyển động môi,...Như nghiên cứu của Jung và Cộng sự [7] đã xây dựng mô hình dựa trên tần suất chớp mắt đạt độ chính xác 87.5%, nghiên cứu của Matern và Cộng sự [14] sử dụng vùng răng và mắt là đặc trưng đạt AUC = 0.851...Tuy nhiên, các phương pháp trên dần kém hiệu quả với các công nghệ tạo Deepfake. Deepfake ngày càng trên nêu tinh vi và rất khó để phân biệt với video thật. Do đó, việc sử dụng các tín hiệu sinh học có thể có hiệu quả hơn so với các phương pháp trên bởi việc làm giả các đặc trưng sinh học vẫn là thách thức đối với các công nghệ Deepfake.

Sử dụng các tín hiệu mang đặc điểm sinh học, đã đem lại những hiệu quả nhất định trong việc nhận dạng video Deepfake. Remote Photoplethysmography (rPPG) đã được như là một đặc trưng sinh học trong nhận dạng video thật/giả. Nhìn chung, các phương pháp đạt được những hiệu quả nhất định. Điểm khác nhau chính giữa các phương pháp sử dụng rPPG ở 2 đặc điểm chính:

- Các phương pháp trích xuất và xử lý tín hiệu sinh học: phương pháp xử lý bằng thuật toán (GREEN CHANNEL, CHROM [4], PCA/ICA, POS [21], LGI [15],...) & phương pháp sử dụng học sâu (DEEPPHYS, TS-CAN [12],...)
- Các mô hình phân loại tín hiệu: sử dụng Support Vector Machine, Convolution neural network hay Recurrent neural network.

Bài báo [5] sử dụng Neural ODE và đặc trưng là nhịp tim trong nhận giả video đạt giá trị lỗi Loss=0.0215. Bài báo FakeCatcher [2] [3] (sử dụng CNN+CHROM) được Intel ứng dụng đã đạt được độ chính xác 96% . Bài báo khác "DeepFakes Have no Heart: a Simple rPPG-based

"Method to Reveal Fake Videos" cũng trích xuất các đặc trưng ước lượng nhịp tim, sau đó, đưa qua mạng SVM (Support Vector Machine), kết quả đạt 96.37% độ chính xác. Bài báo "How Do the Hearts of Deep Fakes Beat? Deep Fake Source Detection via Interpreting Residuals with Biological Signal", sử dụng mạng tích chập CNN và cho ra kết quả đạt 93.69%. Mô hình DeepFakesON-Phys sử dụng mô hình CNN+ Attention mechanism đạt 98.7%. Tuy nhiên đối với mạng RNN, số lượng các nghiên cứu còn hạn chế. Bài báo "Face Biometric Spoof Detection Method Using a Remote Photoplethysmography Signal" [9] sử dụng CHROM + LSTM để nhận dạng người thật/ người đeo mặt nạ để xác thực khuôn mặt. Kết quả đạt 97.68% độ chính xác với tập dữ liệu của tác giả.

Tóm lại, các mô hình sử dụng rPPG đã chứng minh được tính hiệu quả với độ chính xác cao. Thậm chí, những mô hình này còn đạt độ chính xác tốt hơn những mô hình chỉ dựa trên bề ngoài ảnh/ video. Do đó, đây cũng là lý do đồ án thực hiện khảo sát loại tín hiệu này.

2.4 Kết luận chương

Qua chương này, ta đã tìm hiểu tổng quan về các phương pháp trích xuất rPPG (xử lý ảnh truyền thống và sử dụng mô học học sâu), các phương pháp nhận dạng video thật/ giả. Từ cơ sở đó, ta nhận thấy rằng các phương pháp sử dụng CNN phổ biến hơn các phương pháp khác, trong khi đó, phương pháp sử dụng RNN vẫn còn ít nghiên cứu. Do đó, đồ án này sẽ nghiên cứu về độ hiệu quả của mô hình sử dụng RNN. Chương sau sẽ mô tả chi tiết lý thuyết về rPPG, các mô hình và phương pháp đề xuất.

Chương 3

BIẾN ĐỔI FOURIER VÀ BỘ LỌC

Chương 3 sẽ trình bày về các khái niệm cơ bản trong xử lý số tín hiệu: phép biến Fourier, mật độ phổ năng lượng/ phổ công suất, và các bộ lọc tín hiệu. Các bộ lọc tín hiệu được sử dụng trong việc lọc tín hiệu trong miền tần số phù hợp với nhịp tim, trong khi đó, biến đổi Fourier được dùng để biểu diễn tín hiệu trong miền tần số và được sử dụng như là đặc trưng trong quá trình huấn luyện mô hình.

3.1 Biến đổi Fourier rời rạc

Biến đổi Fourier là một phép biến đổi tín hiệu trong miền thời gian sang miền tần số. Dựa vào phép biến đổi này, ta có thể phân tích năng lượng, độ lớn, pha của tín hiệu trong miền tần số. **Biến đổi Fourier của tín hiệu rời rạc** (Discrete Time Fourier transform, viết tắt: DTFT) là phép biến đổi chuyển tín hiệu rời rạc trong miền thời gian $x(n)$ sang miền tần số. Kết quả của biến đổi ta thu được $X(\omega)$, là một hàm liên tục và được tính bởi biểu thức sau:

$$X(\omega) = \sum_{n=-\infty}^{+\infty} x(n)e^{-j\omega n} \quad (3.1)$$

Trong Biểu thức (3.1), $x(n)$ là tín hiệu rời rạc trong miền thời gian, $\omega = \frac{2\pi f}{f_s}$, f_s (Hz) là tần số lấy mẫu, n là chỉ số của tín hiệu và có khoảng giá trị trong khoảng $(-\infty, +\infty)$. Vì vậy, chiều dài của tín hiệu trong miền tần số là vô hạn. Do đó, để khảo sát tín hiệu trong miền tần số, ta thường sử dụng các hàm cửa sổ để quan sát trong một khoảng tín hiệu nhất định. Tổng quát, ta có công thức:

$$X_L(\omega) = \frac{1}{2\pi} X(\omega)W(\omega) \quad (3.2)$$

Trong Biểu thức (3.2), $X_L(\omega)$ là tín hiệu trong miền tần số được quan sát thông qua cửa sổ $W(\omega)$ và $X(\omega)$ chính là biến đổi Fourier của tín hiệu rời rạc $x(n)$ và có chiều dài vô hạn. Hàm cửa sổ thông thường được sử dụng là Hamming, Hanning, Bartlett, Blackman,... Để biến đổi

tín hiệu trong miền tần số về miền thời gian, ta có biểu thức biến đổi ngược như sau:

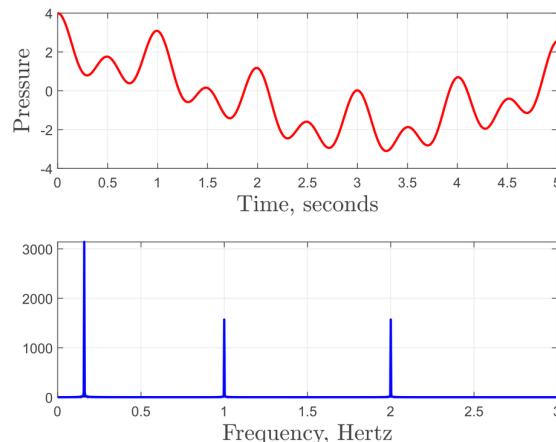
$$x(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(\omega) e^{j\omega n} d\omega \quad (3.3)$$

Trong Biểu thức (3.3), $X(\omega)$ là tín hiệu trong miền tần số, ω là tần số góc của tín hiệu (rad), $x(n)$ là tín hiệu ban đầu.

Biến đổi Fourier rời rạc (Discrete Fourier transform, viết tắt: DFT) chính là quá trình rời rạc hóa $X(\omega)$. Biểu thức tính toán như sau:

$$X(k) = \sum_{n=0}^{L-1} x(n) e^{-j\frac{2\pi kn}{N}}, k = 0, \dots, N-1 \quad (3.4)$$

Trong đó, $X(k)$ là tín hiệu sau khi biến đổi Fourier rời rạc. Biểu thức (3.4) có độ dài L mẫu cho thấy phỏ tần số $X(\omega)$ được lấy mẫu tại N tần số ($N \geq L$) cách đều nhau một khoảng $\omega = \frac{2\pi k}{N}$. Thông thường, $N = L$: số lượng mẫu được lấy mẫu bằng với số lượng tần số cần lấy mẫu. Hình 3.1 sẽ minh họa cho biến đổi Fourier và tín hiệu trong miền tần số.



Hình 3.1: Tín hiệu trong miền thời gian và miền tần số

Tương tự như phân trên, biến đổi DFT cũng có phép biến đổi ngược gọi là IDFT, được tính như sau:

$$x(n) = \frac{1}{N} \sum_{n=0}^{L-1} X(k) e^{-j \frac{2\pi kn}{N}}, k = 0, \dots, N-1 \quad (3.5)$$

Trong Biểu thức (3.5), $x(n)$ là tín hiệu rời rạc trong miền thời gian, $X(k)$ là tín hiệu đã được biến đổi Fourier rời rạc trong miền tần số, N là số mẫu tần số được lấy mẫu, L là độ dài của tín hiệu.

Như vậy, phần trên đã khảo sát một số kỹ thuật biến đổi Fourier đối với tín hiệu rời rạc. Dựa trên biến đổi Fourier, ta có thể khảo sát được năng lượng, phổ tần số. Phần tiếp theo sẽ thể hiện rõ hơn vai trò này của biến đổi Fourier.

3.2 Mật độ phổ

Mật độ phổ năng lượng (ESD) được dùng để mô tả phân bố năng lượng theo tần số của tín hiệu. Theo Định lý Parseval, ta có công thức tính mật độ phổ năng lượng như sau:

$$E = \sum_{n=-\infty}^{+\infty} |x(n)|^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} |X(\omega)|^2 d\omega \quad (3.6)$$

Trong đó, E là mật độ phổ năng lượng, $x(n)$ và $X(\omega)$ lần lượt là tín hiệu rời rạc trong miền thời gian và biến đổi Fourier trong miền tần số.

Mật độ phổ công suất (PSD) là đại lượng được sử dụng phổ biến hơn ESD và được tính bằng cách lấy mật độ phổ năng lượng chia cho chu kỳ lấy mẫu của tín hiệu:

$$P = \frac{E}{T} \quad (3.7)$$

Với P là mật độ phổ công suất, E là mật độ phổ năng lượng và T là chu kỳ của tín hiệu.

3.3 Bộ lọc tín hiệu

Trong phần này, sẽ trình bày các bộ lọc được sử dụng trong xử lý số tín hiệu. Gồm có 4 loại bộ lọc chính: bộ lọc thông thấp, bộ lọc thông cao, bộ lọc thông dải, bộ lọc chấn dải. Cụ thể: bộ lọc thông thấp là bộ lọc chỉ giữ lại các tín hiệu có thành phần tần số trong khoảng $f \leq f_{max}$. Bộ lọc thông cao là bộ lọc chỉ giữ lại các tín hiệu có thành phần tần số trong khoảng $f \geq f_{min}$. Bộ lọc thông dải là bộ lọc giữ lại các tín hiệu có thành phần tần số trong khoảng $f_{min} \leq f \leq f_{max}$. Và cuối cùng, bộ lọc chấn dải là bộ lọc giữ lại các tín hiệu có thành phần tần số nằm ngoài khoảng $f_{min} \leq f \leq f_{max}$.

Dưới đây là một số loại bộ lọc: bộ lọc FIR và bộ lọc IIR.

- **Bộ lọc FIR** (finite impulse response):

Là bộ lọc có đáp ứng xung có chiều dài hữu hạn, với $0 \leq n \leq M$ đặc trưng bởi biểu thức:

$$y(n) = \sum_{m=0}^M h(m) * x(n-m) \quad (3.8)$$

Với $y(n)$ là tín hiệu thu được sau khi lọc, M được gọi là bậc bộ lọc, $h(m)$ là đáp ứng xung của bộ lọc, $x(n-m)$ là tín hiệu được dời trong miền thời gian.

- **Bộ lọc IIR** (infinite impulse response):

Là bộ lọc có đáp ứng xung dài vô hạn, đặc trưng bởi biểu thức:

$$y(n) = \sum_{m=0}^{\infty} h(m) * x(n-m) \quad (3.9)$$

Về cơ bản, Biểu thức (3.9) giống với Biểu thức (3.8). Điểm khác biệt giữa hai biểu thức đến từ độ dài của bộ lọc.

Có rất nhiều phương pháp khác nhau để thiết kế bộ lọc như Butterworth, Chebyshev, Elliptic,... Trong khuôn khổ đồ án này, bộ lọc Butterworth sẽ được xem xét và trình bày cụ thể bên dưới. Đáp ứng biên độ của bộ lọc thông thấp Butterworth bậc n có công thức như sau:

$$|H(j\omega)| = \frac{1}{\sqrt{1 + (\frac{\omega}{\omega_c})^{2n}}} \quad (3.10)$$

Trong Biểu thức (3.10), ω_c là tần số cắt. Tại tần số cắt ω_c , đáp ứng biên độ suy hao -3dB so với biên độ cực đại. Trong thiết kế, ta thường dùng đáp ứng chuẩn hóa ($\omega_c=1$), khi này biểu thức trở thành:

$$|H(j\omega)_{LP}| = \frac{1}{\sqrt{1 + (\omega)^{2n}}} \quad (3.11)$$

Thực hiện biến đổi Laplace, ta có được biểu thức:

$$H(s) = \frac{1}{(s - s_1)(s - s_1)(s - s_2)...(s - s_n)} \quad (3.12)$$

Bộ lọc Butterworth được sử dụng nhiều trong các bộ lọc tương tự (analog). Do đó, ta cần thực hiện một vài phép biến đổi để có thể sử dụng được với tín hiệu số (digital). Biến đổi Bilinear thể hiện mối quan hệ giữa biến đổi Laplace và biến đổi Z, đặc trưng bởi biểu thức sau:

$$s = \frac{2}{T} \left(\frac{1 - z^{-1}}{1 + z^{-1}} \right) \quad (3.13)$$

Trong đó, $T(s)$ là thời gian lấy mẫu. Thực hiện ánh xạ phi tuyến chuyển Ω sang ω . Biểu thức xác định như sau:

$$\Omega = \frac{2}{T} \tan\left(\frac{\omega}{2}\right) \quad (3.14)$$

$$\omega = 2 \arctan\left(\frac{\Omega T}{2}\right) \quad (3.15)$$

Dể thuận tiện cho việc tính toán, ta xây dựng bảng giá trị chuẩn hóa $\omega_c = 1$ với các bậc n như ở Bảng (3.3):

Bảng 3.1: Biểu thức tương ứng với các bậc

Bậc(n)	Denominator Polynomial
1	$s+1$
2	$s^2+1.414s+1$
3	$(s+1)(s^2+s+1)$
4	$(s^2+0.766s+1)(s^2+1.848s+1)$
5	$(s+1)(s^2+0.618s+1)(s^2+1.618s+1)$
6	$(s^2+0.518s+1)(s^2+1.414s+1)(s^2+1.932s+1)$
7	$(s+1)(s^2+0.445s+1)(s^2+1.802s+1)$

Như vậy, khi thực hiện việc thiết kế bộ lọc Butterworth, ta sẽ thực hiện việc chuẩn hóa tần số và thiết kế dựa trên Bảng (3.3). Sau đó, ta sẽ chuyển lại về miền tần số ban đầu. Tương tự với bộ lọc thông cao:

$$|H(j\omega)_{HP}| = 1 - |H(j\omega)_{LP}| \quad (3.16)$$

Bộ lọc thông dải, ta có biểu thức:

$$|H(j\omega)_{BP}| = |H(j(\omega - \omega_0))_{LP}| \quad (3.17)$$

Với Biểu thức (3.17), ω_0 là giá trị dùng để dịch chuyển phô tần số của bộ lọc thông thấp. Cuối cùng, ta có bộ lọc chấn dải có biểu thức như sau:

$$|H(j\omega)_{SP}| = 1 - |H(j\omega)_{BP}| \quad (3.18)$$

Trong các Biểu thức (3.16), (3.17), (3.18): ký hiệu $|H(j\omega)|$ chỉ độ lớn đáp ứng xung, LP, HP, BP, SP ứng với bộ lọc thông thấp, thông cao, thông dải và chấn dải.

Trong đồ án này, các bộ lọc được sử dụng để lọc tín hiệu rPPG trong miền tần số với mục tiêu loại bỏ nhiễu.

3.4 Kết luận chương

Chương này đã khái quát về biến đổi Fourier, một biến đổi quan trọng và cơ bản trong xử lý tín hiệu dùng để phân tích tín hiệu trong miền tần số. Chương này cũng giới thiệu về bộ lọc

được ứng dụng để thu nhập tín hiệu trong miền tần số mong muốn. Chương tiếp theo sẽ trình bày về Deepfake và tổng quan các nghiên cứu trong việc phát hiện video giả.

Chương 4

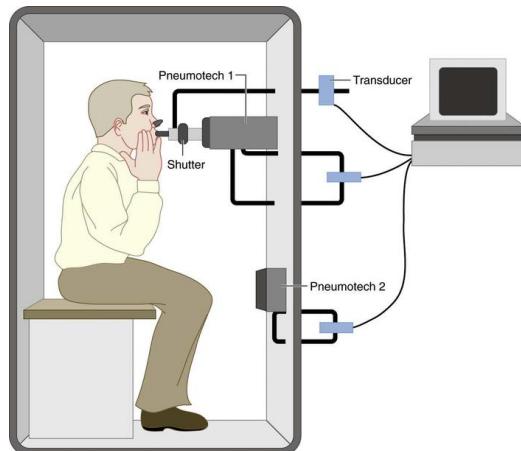
CƠ SỞ SINH HỌC VÀ KỸ THUẬT TRÍCH XUẤT REMOTE PHOTOPLETHYSMOGRAPHY (RPPG)

Như đã trình bày ở chương trước, các tín hiệu sinh học như rPPG có hiệu quả nhất định trong việc phân biệt video thật và giả. Tín hiệu rPPG về bản chất có nguồn gốc từ việc thay đổi ánh sáng do lượng máu lưu thông gây ra. Chương này sẽ trình bày cơ sở sinh học và từ đó, tìm hiểu các phương pháp để thu nhận được tín hiệu rPPG. Và chương cũng thực hiện việc so sánh hiệu quả của những phương pháp thông qua các nghiên cứu trước đó.

4.1 Cơ sở sinh học

Nhịp tim (Heart rate) là một trong những thông tin sinh học quan trọng đối với con người. Một trong những cách hiệu quả để đo lường nhịp tim dựa vào việc đo điện tâm đồ (ECG). Một phương pháp khác được sử dụng đo lường nhịp tim sử dụng Plethysmography. Đây là một phương pháp được sử dụng trong y khoa để đo sự thay đổi thể tích trong một cơ quan hay toàn bộ cơ thể (thể tích máu hoặc khí). Kỹ thuật có nhiều ứng dụng như: Body plethysmography (Do thể tích khí trong phổi), Pulse plethysmography (Do lưu lượng máu và nhịp tim),...Dưới đây là hình ảnh minh họa cho ứng dụng của Plethysmography được ứng dụng trong y tế:

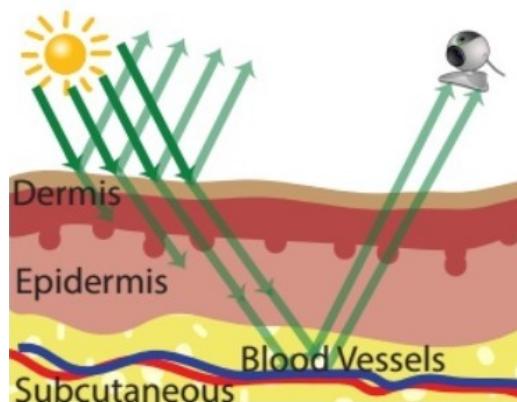
Hình 4.1 cho thấy rằng để đo thể tích phổi người bệnh được yêu cầu thở vào thiết bị để đo lường. Ngày nay, công nghệ Plethysmography được ứng dụng nhiều hơn trong việc đo lường nhịp tim dưới dạng sử dụng sự hấp thụ của ánh sáng; kỹ thuật này được gọi là Photoplethysmography. Thiết bị sẽ phát ra một chùm ánh sáng và khi ánh sáng đó đến vùng da, một số tia sáng sẽ bị hấp thụ, số khác sẽ phản xạ lại thiết bị. Theo thời gian, mức độ phản xạ ánh sáng sẽ khác nhau do sự thay đổi của lưu lượng máu.



Hình 4.1: Body Plethysmography

Từ đây các thiết bị sẽ thực hiện tính toán nhịp tim. Kỹ thuật này được ứng dụng rộng rãi trong các thiết bị di động như đồng hồ thông minh, điện thoại, máy đo SPO₂,...

Dể miêu tả rõ hơn dưới góc độ tế bào sinh học, ta có mô hình sau:



Hình 4.2: Sự hấp thụ và phản xạ của tế bào

Hình 4.2 minh họa cho cách đo lường Photoplethysmography. Khi tim hoạt động, lưu lượng máu sẽ truyền đi với mức độ khác nhau. Khi ánh sáng xung quanh chiếu vào một khu vực trong cơ thể, lượng áng sáng hấp thụ và phản xạ cũng sẽ khác nhau (do sự thay đổi lưu lượng máu). Dựa vào việc thu thập sự thay đổi ánh sáng này, người ta ước lượng nhịp tim. Tuy nhiên phương pháp này tốn kém và yêu cầu việc tiếp xúc trực tiếp.

Trong một số trường hợp việc tiếp xúc trực tiếp không phù hợp. Cũng dựa trên ý tưởng đo lường sự thay đổi ánh sáng, kỹ thuật Remote Photoplethysmography (rPPG) đã ra đời. Kỹ thuật này sử dụng các video làm dữ liệu đầu vào để đo lường sự thay đổi mức xám (ánh sáng). Tuy nhiên, vấn đề chính xác của rPPG còn nhiều thách thức. Do đó, thông thường các tín hiệu phải thực hiện xử lý qua các bộ lọc tín hiệu và các kĩ thuật thống kê khác nhau để đạt được kết quả tốt nhất.

Tóm lại, việc đo lường giá trị nhịp tim dựa vào kĩ thuật remote photoplethysmography chính là việc đo lường sự thay đổi mức xám của các pixel trong miền tần số thấp, nguyên nhân của sự

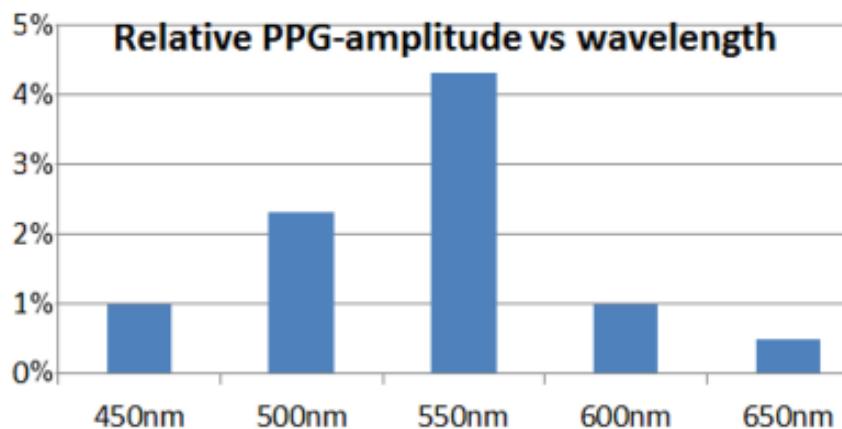
thay đổi này do sự co bóp của tim dẫn đến sự thay đổi lưu lượng máu tại các vị trí trên cơ thể.

4.2 Thuật toán

Ở phần này sẽ khảo sát về các thuật toán chính được sử dụng trong trích xuất tín hiệu rPPG. Nhìn chung, có 2 phương pháp tiếp cận: sử dụng các mô hình xử lý ảnh và sử dụng các mô hình học sâu (deep learning). Một số mô hình đại diện cho phương pháp truyền thống bao gồm: Green channel based rPPG, CHROM, POS, PCA/ ICA,... Các phương pháp đại diện cho mô hình học sâu bao gồm: DeepPhys, MTTS CAN,... Tuy nhiên, trong khuôn khổ đồ án này, đồ án chỉ khảo sát đến các phương pháp truyền thống được ứng dụng trong việc xây dựng mô hình ở Chương 5.

4.2.1 Green channel based rPPG

Như đã trình bày phía trên, Remote Plethysmography là kỹ thuật dựa trên sự thay đổi của mức xám của các pixel. Tuy nhiên, những sự thay đổi này là rất nhỏ. Để thu được kết quả tốt nhất, [4] đã thực hiện khảo sát sự hấp thụ và phản xạ ánh sáng của các vùng trên da. Kết quả được thể hiện ở hình bên dưới:



Hình 4.3: Khảo sát mức độ hấp thụ và phản xạ ánh sáng

Như Hình 4.3, ánh sáng có bước sóng = 550nm cho độ lớn rPPG lớn nhất. Bước sóng này tương ứng với màu xanh lá. Dựa trên ý tưởng đó, nghiên cứu [19] đã sử dụng kênh màu xanh lá (hệ màu RGB) để trích xuất rPPG. Đây được xem là một trong những phương pháp đầu tiên trong việc ước lượng rPPG.

$$S(t) = \frac{1}{N} \sum_{i=1}^N g_i(t) \quad (4.1)$$

Với $S(t)$ là tín hiệu đại diện cho rPPG tại thời điểm t được tính bằng cách lấy trung bình

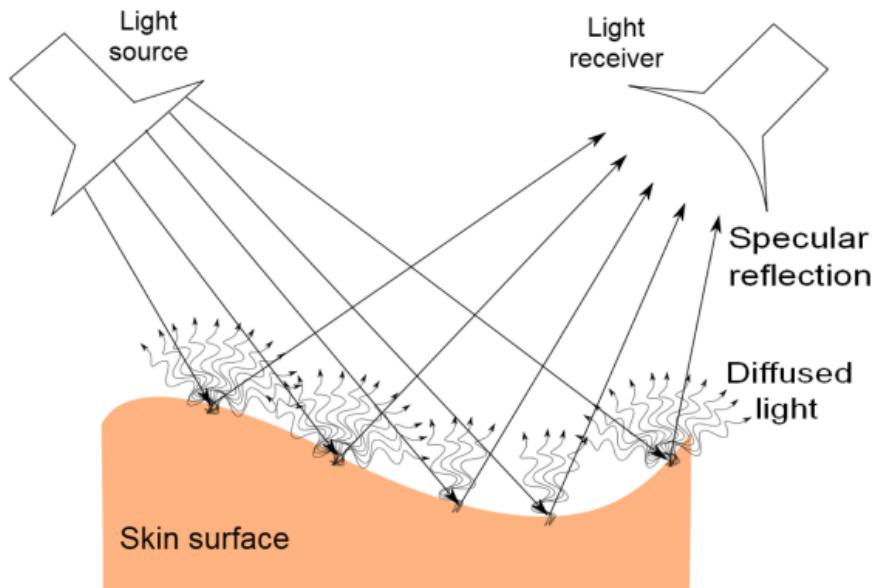
của $g_i(t)$, đây là giá trị mức xám của kênh màu xanh lá tại pixel thứ i . N là tổng số lượng điểm ảnh (pixel) có trong vùng ảnh đang xét.

Tín hiệu nhịp tim thông thường trong khoảng từ 30 BPM (Beats per minute) đến 200 BPM, ứng với tần số 0.5-3.3 Hz. Do đó, bước cuối cùng để thu được giá trị nhịp tim, ta thường sử dụng bộ lọc thông dải đối với tín hiệu rPPG thô để thu được miền tần số trên.

Dây là một trong những phương pháp đầu tiên được sử dụng để trích xuất rPPG. Tuy nhiên, phương pháp này khá đơn giản và chưa xem xét đến các yếu tố nhiễu. Do đó, phương pháp này thường không được sử dụng các ứng dụng thực tế. Phương pháp là tiền đề của các phương pháp khác như PCA/ICA...

4.2.2 Chrominance based rPPG

Phương pháp trích xuất rPPG từ kênh Green là một phương pháp cổ điển, do đó, phương pháp này còn có những hạn chế do bị ảnh hưởng bởi độ rời sáng (illuminance). Do đó, phương pháp dựa trên Chrominance nhằm giảm ảnh hưởng của chuyển động và độ rời sáng illuminance lên tín hiệu rPPG. Trước khi hiểu về cơ chế của CHROM, ta xem xét mô hình sau:



Hình 4.4: Mô hình phản xạ

Trong Hình 4.4, khi một chùm tia sáng (Light source) đến bề mặt của da người (Skin surface), một phần ánh sáng đó sẽ được khuếch tán trên khắp bề mặt da và phản xạ lại một phần (Diffused reflection), phần còn lại sẽ phản xạ lại trực tiếp thiết bị (Specular reflection) thu nhận ánh sáng (Light receiver), trong trường hợp này là camera. Thành phần phản xạ khuếch tán thay đổi (Diffused reflection) do sự co bóp của nhịp tim, điều này dẫn đến sự thay đổi của cường độ ánh sáng. Về mặt toán học, cường độ của một điểm ảnh (pixel) i ở kênh màu $C \in \{R, G, B\}$ (C là một trong ba kênh màu Red (R), Green (G) và Blue (B)) được biểu diễn

như sau:

$$C_i(t) = I_{Ci}(t)(\rho_{Cdc} + \rho_{Ci}(t) + s_i(t)) \quad (4.2)$$

Trong đó $I_{Ci}(t)$ là cường độ ánh sáng do máy ảnh thu được. Để đặc trưng cho phản xạ khuếch tán (Diffused reflection), ta dùng đại lường ρ gồm hai thành phần ρ_{Cdc} là thành phần cố định của hệ số phản xạ của da, $\rho_{Ci}(t)$ là thành phần biến thiên với giá trị trung bình do sự dao động lượng máu gây ra. $s_i(t)$ là thành phần đóng góp cho phản xạ phản chiếu (Specular reflection).

Để đơn giản hóa biểu thức, yếu tố thời gian t sẽ được ngầm hiểu. Khi này biểu thức sẽ thành:

$$C_i = I_{Ci}(\rho_{Cdc} + \rho_{Ci} + s_i) \quad (4.3)$$

Mục tiêu của thuật toán là trích xuất thành phần ρ_{Ci} bằng loại bỏ thành phần I_{Ci} , ρ_{Cdc} và s_i . Để thực hiện mục tiêu này, ta thực hiện lần lượt các bước sau. Đầu tiên, ta thực hiện việc chuẩn hóa từng kênh màu. Công thức chuẩn hóa như sau:

$$C_{ni} = \frac{C_i}{\mu(C_i)} \quad (4.4)$$

Trong Biểu thức (4.4), C_{ni} là giá trị của kênh màu sau khi chuẩn hóa, C_i là giá trị ban đầu và $\mu(C_i)$ là giá trị trung bình tổng số N pixel, và được tính bằng biểu thức:

$$\mu(C_i) = \frac{1}{N} \sum_{i=0}^N (C_i) \quad (4.5)$$

Giá trị ρ_{Cdc} giữ lại thông tin chung của tín hiệu, tức thành phần DC. Do đó, việc chuẩn hóa C_{ni} nhằm giảm hạn chế của thành phần ρ_{Cdc} trong Biểu thức (4.2).

Trong quá trình đo lường khó tránh được những chuyển động của đầu (head movement). Các chuyển động này làm thay đổi I_{Ci} , khi này, việc chuẩn hóa tín hiệu thường như không còn hiệu quả. Tuy nhiên, đối với hệ màu RGB, hệ số I_{Ci} gần như bằng nhau ở các kênh màu. Do đó, việc sử dụng tỉ lệ giữa các kênh màu có thể giải quyết vấn đề trên. Vậy nên, ta có biểu thức sau:

$$S_i = \frac{G_{ni}}{R_{ni}} - 1 \quad (4.6)$$

Trong Biểu thức (4.6), S_i đại diện cho tín hiệu rPPG, G_{ni} , R_{ni} đại diện cho kênh màu xanh lá và đỏ đã được chuẩn hóa. Vấn đề còn lại ở đại lượng s_i của Biểu thức (4.6). Do đó, ta thực hiện chuyển kênh màu RGB sang hệ X,Y. Một cách tổng quát ta có:

$$X = R - G \quad (4.7)$$

$$Y = 0.5R + 0.5G - B \quad (4.8)$$

Tác giả cho rằng việc chuẩn hóa (standardization) giá trị R,G,B sẽ đem lại kết quả tốt hơn. Dựa vào các thử nghiệm, tác giả chuẩn hóa các kênh màu như sau:

$$R_s = 0.7682R_n \quad (4.9)$$

$$G_s = 0.5121G_n \quad (4.10)$$

$$B_s = 0.3841B_n \quad (4.11)$$

Như vậy, cuối cùng ta được biểu thức:

$$S_i = \frac{X_s}{Y_s} - 1 \quad (4.12)$$

Với X_s, Y_s là các giá trị tính theo Biểu thức (4.7), (4.8) với R,G,B là các giá trị được chuẩn hóa.

Sử dụng phân tích logarit và chuỗi Taylor, cuối cùng ta có biểu thức xấp xỉ sau:

$$S_i \approx X_s - Y_s = 1.5R - 3G + 1.5B \quad (4.13)$$

Để cải thiện độ chính xác, ta thêm thông số tinh chỉnh α và thực hiện lọc tín hiệu trong khoảng tần số của nhịp tim, biểu thức trở thành:

$$S_i = X_f - \alpha Y_f \quad (4.14)$$

Với X_f, Y_f là tín hiệu được lọc thông dải trong khoảng tần số 0.7 đến 2.5Hz, ứng với tần số của nhịp tim. Với hệ số α được tính theo biểu thức:

$$\sigma(Y_f \cdot \alpha) = \sigma(Y_f) \rightarrow \alpha = \frac{\sigma(X_f)}{\sigma(Y_f)} \quad (4.15)$$

Ký hiệu σ biểu thị cho phép tính phương sai. Do đó, mục đích của việc tính α để tinh chỉnh giá trị của X_f và Y_f .

Cuối cùng, thay Biểu thức (4.7) và (4.8) vào (4.14), biểu thức trở thành:

$$S = 3\left(1 - \frac{\alpha}{2}\right)R_f - 2\left(1 - \frac{\alpha}{2}\right)G_f + \frac{3\alpha}{2}B_f \quad (4.16)$$

Với R_f, G_f, B_f là các tín hiệu đã được lọc thông dải.

Tóm lại, phương pháp CHROM đã khắc phục những khuyết điểm của phương pháp CHROM nhờ vào việc loại bỏ các thành phần độ rời (illuminance). Việc này giúp cải thiện độ chính xác của phép đo do giảm đi ảnh hưởng của màu da và chuyển động đầu.

4.2.3 Plane-Orthogonal-to-Skin (POS)

Dựa trên Biểu thức (4.2), POS thực hiện phân tích s_i của phản xạ specular và thành phần $I_{Ci}(t)$ thành hai thành phần cố định và thay đổi. Như vậy, biểu thức sẽ trở thành

$$C_i(t) = I_0(1 + i(t))(\rho_{Cdc} + \rho_{Ci}(t) + s_i(t) + s_0) \quad (4.17)$$

Với I_0 là cường độ ánh sáng không thay đổi, $i(t)$ là thành phần làm thay đổi cường độ ánh sáng của máy ảnh. Ta mong muốn biểu diễn biểu thức dưới dạng vector. Gọi u_s , u_d và u_p lần lượt vector đơn vị của thành phần phản xạ Specular $s_i(t)$, vector đơn vị của thành phần ρ_{Cdc} , vector đơn vị của thành phần $\rho_{Ci}(t)$. Cụ thể, ta có biểu thức như sau:

$$\rho_{Cdc}(t) = u_d \cdot d_0 \quad (4.18)$$

$$\rho_{Ci} = u_p \cdot p(t) \quad (4.19)$$

$$s_0 + s_i(t) = u_s \cdot (s_0 + s_i(t)) \quad (4.20)$$

Trong đó, d_0 là các giá trị độ lớn của ρ_{Cdc} , $(s_0 + s_i(t))$ đại diện cho tổng độ lớn của thành phần cố định, thay đổi của phản xạ specular và $p(t)$ là giá trị độ lớn của rPPG theo thời điểm t .

Như vậy, biểu diễn dưới dạng vector Biểu thức (4.17) sẽ trở thành:

$$C_i(t) = I_0(1 + i(t))(u_d \cdot d_0 + u_p \cdot p(t) + u_s \cdot (s_0 + s_i(t))) \quad (4.21)$$

Đặt $u_c \cdot c_0 = u_s \cdot s_0 + u_d \cdot d_0$ và thay vào Biểu thức (4.21), ta được:

$$C_i(t) = I_0(1 + i(t))(u_c \cdot c_0 + u_s \cdot s(t) + u_p \cdot p(t)) \quad (4.22)$$

Ta thực hiện chuẩn hóa $C_i(t)$ bằng cách nhân với ma trận N như sau:

$$C_n(t) = N \cdot C_i(t) \quad (4.23)$$

Với $C_n(t)$ là giá trị C_i đã được chuẩn hóa. N là ma trận thỏa điều kiện sau:

$$N \cdot \mu(C_i(t)) = 1 \quad (4.24)$$

Với $\mu(C_i(t))$ là giá trị trung bình của $C_i(t)$ theo thời gian t .

Thực hiện thay thế các biểu thức, cuối cùng ta được biểu thức:

$$C_n(t) = 1.(1 + i(t)) + N.u_s.I_0.s(t) + N.u_p.I_0.p(t) \quad (4.25)$$

Trong đó:

- $1.(1 + i(t))$ đại diện cho cường độ sáng (intensity)
- $N.u_s.I_0.s(t)$ đại diện cho specular reflection
- $N.u_p.I_0.p(t)$ đại diện sự thay đổi ánh sáng do nhịp tim gây ra.

Dựa trên mô hình trên, ta có thể biểu diễn tín hiệu rPPG dưới dạng phép chiếu. Ta có được biểu thức sau:

$$S(t) = P_p.C_n(t) \quad (4.26)$$

Với P_p là ma trận chiếu có kích thước 2×3 thoải điều kiện sau:

$$P_p \cdot 1 = (0, 0)^T \quad (4.27)$$

$$P_{p,1} \cdot P_{p,2}^T = 0 \quad (4.28)$$

Dựa vào các thử nghiệm của tác giả [20], ta có được ma trận chiếu P_p được định nghĩa như sau:

$$P_p = \begin{bmatrix} 0 & 1 & -1 \\ -2 & 1 & 1 \end{bmatrix} \quad (4.29)$$

Khi này $S_1(t)$ và $S_2(t)$ trở thành như sau:

$$S_1(t) = G_n(t) - B_n(t) \quad (4.30)$$

$$S_2(t) = G_n(t) + B_n(t) - 2R_n(t) \quad (4.31)$$

Ta sử dụng kĩ thuật alpha-tuning, cuối cùng ta thu được biểu thức sau:

$$h(t) = S_1(t) + \alpha \cdot S_2(t) \quad (4.32)$$

Với $h(t)$ là giá trị ngõ ra tín hiệu rPPG và hệ số α được tính theo công thức sau:

$$\alpha = \sigma(S_1)/\sigma(S_2) \quad (4.33)$$

Tương tự với phương pháp CHROM, α cũng được sử dụng trong việc tinh chỉnh giá trị của S_1 và S_2 . Trong trường hợp này dấu "+" của Biểu thức (4.32), thể hiện rằng S_1 và S_2 cùng pha với nhau.

Như vậy, có thể xem POS là một biến thể của phương pháp CHROM và theo các thử nghiệm ([20]), đã cho thấy POS có kết quả tốt hơn CHROM. Tuy nhiên, nhìn chung cả CHROM và

POS tập trung giải quyết vấn đề khác biệt do màu da và độ rọi. Phương pháp LGI bên dưới sẽ tập trung nhiều hơn về các ảnh hưởng của chuyển động đầu.

4.2.4 Local group invariance (LGI)

LGI sử dụng các thành phần bất biến (invariance) cho việc ước lượng nhịp tim. Vấn đề "tính không thay đổi" (invariance problem) liên quan đến việc thiết kế các mô hình hoặc thuật toán học máy có khả năng xử lý các loại biến đổi hoặc biến thể khác nhau trong dữ liệu đầu vào. Trong nhiều ứng dụng thực tế, dữ liệu đầu vào có thể trải qua các biến đổi khác nhau (ví dụ: dịch chuyển, xoay, tỉ lệ) mà không thay đổi các mẫu hoặc nhãn mục tiêu cơ bản. Mục tiêu là xây dựng các mô hình có thể nhận diện mẫu hoặc dự đoán đúng kết quả bất kể các biến đổi này. Khi đo lường rPPG, tín hiệu có thể bị sai lệch do góc quay, vị trí mặt,... Dựa trên ý tưởng đó, thuật toán LGI mong muốn giải quyết vấn đề sai lệch do di chuyển đầu, bằng cách giữ lại các thành phần bất biến (nhịp tim). Đầu tiên, ta sẽ tìm hiểu một số khái niệm sau:

Ma trận hiệp phương sai (Covariance Matrix)

Ma trận hiệp phương sai của của X gồm n vector $X = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]$ với mỗi vector gồm m chiều được tính bởi biểu thức:

$$C = \frac{1}{N} \sum_{i=1}^N (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T \quad (4.34)$$

Trong Biểu thức (4.34), ma trận hiệp phương sai là ma trận vuông với kích thước $m \times m$. \mathbf{x} đại diện cho vector thứ i trong X, $\bar{\mathbf{x}}$ đại diện cho giá trị trung bình của X.

Biểu diễn dưới dạng ma trận, ta có biểu thức:

$$C = \frac{1}{N} \cdot \hat{X} \cdot \hat{X}^T \quad (4.35)$$

Với \hat{X} là ma trận chứa các phần tử $\hat{x}_i = x_i - \bar{x}$

Từ đó, ta rút ra nhận xét rằng: ma trận hiệp phương sai thể hiện mối quan hệ cùng chiều hoặc nghịch chiều giữa hai biến, sự biến động của các biến. Từ đó, cho ta biết được mối quan hệ giữa các thành phần trong ma trận.

Trị riêng và vector riêng

Giá trị λ được gọi là trị riêng của ma trận A, nếu tồn tại một vector X_0 khác không sao cho:

$$A \cdot X_0 = \lambda \cdot X_0 \quad (4.36)$$

Trong Biểu thức (4.36), X_0 chính là vector riêng, λ là trị riêng tương ứng với vector riêng X_0 . Dựa vào biểu thức (4.36), nếu trong không gian 2 chiều, ta nhận thấy rằng vector $A \cdot X_0$ sẽ

có cùng phương và cùng chiều với vector riêng X_0 nhưng khác nhau về mặt độ lớn. Trị riêng và vector riêng được ứng dụng trong nhiều mục đích khác nhau.

Phân tích suy biến

Phép phân tích suy biến (Singular Value Decomposition) là phép phân tích ma trận thành các ma trận con, được cho bởi biểu thức sau:

$$A = U \sum V^T \quad (4.37)$$

Trong Biểu thức (4.37), A là ma trận đầu vào có kích thước m x n, U và V lần lượt là ma trận trực giao, \sum là ma trận đường chéo có dạng

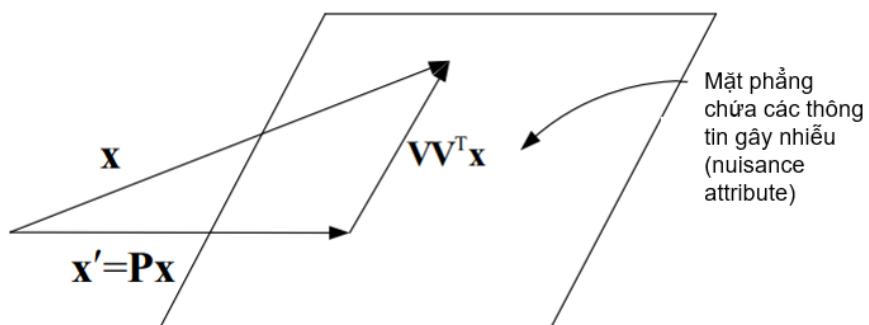
$$\begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix} \quad (4.38)$$

Với D là ma trận chéo gồm các phần tử trên đường chéo là những số thực dương. Đây là các singular values của vector A. Phân tích SVD thường được sử dụng trong bài toán tìm trị riêng và vector riêng của ma trận. Xét một bài toán trị riêng như sau ở Biểu thức (4.36). Ta thực hiện việc chuyển về và có được biểu thức sau:

$$A = X_0 \cdot \lambda \cdot X_0^T \quad (4.39)$$

Như vậy Biểu thức (4.39), cho thấy được biểu thức cũng có dạng là các ma trận được phân ra, điều này tương tự với phương pháp phân tích SVD. Điều này cho thấy bản chất bài toán tìm trị riêng, vector riêng chính là phân tích ma trận thành các ma trận con nhân với nhau.

Quay với với thuật toán LGI, mục tiêu của thuật toán là tìm một mặt phẳng chiếu sao cho những sự biến đổi do chuyển động sẽ không ảnh hưởng đến tín hiệu rPPG. Xét một tín hiệu $\vec{x} = [R, G, B]$ chứa giá trị trung bình của 3 kênh màu Red, Green và Blue. Ta mong muốn tìm được một mặt phẳng P như hình:



Hình 4.5: Minh họa phép chiếu mặt phẳng

Mặt phẳng này thỏa mãn điều kiện $P = I - VV^T$. Với I là ma trận đơn vị. Nhiệm vụ khi này phải tìm V phù hợp với bài toán. Mặt phẳng này phải đặc trưng cho những biến đổi của tín hiệu. Để đặc trưng cho sự thay đổi này, ta sẽ xem xét đến ma trận hiệp phương sai. Trong bài toán này, ma trận hiệp phương sai có dạng:

$$C = \frac{1}{l} \sum_{i=1}^l \left(\frac{\partial}{\partial T} |_{T=0} f(L_T, \vec{x}_i) \right) \left(\frac{\partial}{\partial T} |_{T=0} f(L_T, \vec{x}_i) \right)^T \quad (4.40)$$

Trong Biểu thức (4.40), l là số lượng frame (hay độ dài của tín hiệu). $f(L_T, \vec{x}_i)$ là ngõ ra của tín hiệu bị ảnh hưởng bởi các chuyển động. Trong quá trình đo lường rPPG, phần lớn sự thay đổi diễn ra do sự chuyển động của đầu, cơ thể và những thay đổi này chiếm phần lớn của tín hiệu. Điều này được lý giải do tín hiệu rPPG thường rất nhỏ. Do đó, để giữ lại những biến đổi chính do chuyển động gây ra bằng việc sử dụng trị riêng và vector đặc trưng cho phương sai, trong đó, các vector riêng có giá trị riêng lớn hơn thường chứa nhiều thông tin hơn về biến đổi của dữ liệu. Khi này bài toán trị riêng như sau:

$$CV = V\lambda \quad (4.41)$$

Để giải quyết bài toán này, ta sử dụng SVD (đã được trình bày ở trên) để tìm ra các vector riêng và trị riêng. Như vậy, vector riêng của bài toán này chính là V cần tìm do V mang đặc trưng của sự biến đổi, tức phương sai. Nói cách khác, khi thực hiện phép chiếu mọi biến đổi do chuyển động gây ra sẽ không làm thay đổi đến tín hiệu. Khi này, ta có biểu thức như sau:

$$\vec{x}' = P * \vec{x} \quad (4.42)$$

Nhìn chung, phương pháp LGI thực hiện việc giữ lại các thành phần quan trọng và không thay đổi do chuyển động. Phương pháp ở đây sử dụng các phép chiếu vector để loại bỏ thành phần không mong muốn. Phản tiếp theo sẽ thực hiện việc so sánh giữa các thuật toán với nhau.

4.3 So sánh giữa các thuật toán

Sau khi khảo sát về các lý thuyết của các thuật toán trích xuất rPPG thông dụng, ta có bảng tóm tắt sau:

Các phương pháp cũng có những ưu và nhược điểm riêng. Phương pháp Green channel based [19] bị nhiễu bởi độ rời, nhưng lại không bị ảnh hưởng bởi nén ảnh. Các phương pháp PCA/ICA nói riêng và các kỹ thuật BSS (Blind Source Separation) chỉ xử lý về mặt thống kê của tín hiệu theo miền thời gian mà không quan tâm đến vấn đề tần số. Ngoài ra, phương pháp này còn bị ảnh hưởng bởi độ rời. Phương pháp CHROM [4] và POS [21] khắc phục được vấn đề độ rời những lại nhạy với chuyển động. Ngược lại, LGI [15] chú trọng giải quyết vấn đề chuyển động hơn là độ rời. Trong điều kiện đo lường thực tế, phương pháp LGI và POS cho kết quả tốt

Bảng 4.1: So sánh giữa các phương pháp trích xuất rPPG

Phương pháp	Đặc tính
GREEN	Sử dụng kênh màu xanh lá để trích xuất
ICA / PCA	Sử dụng phương pháp BSS để trích xuất thành phần chính
CHROM	Loại bỏ thành phần độ rời
POS	Sử phép chiếu với mặt phẳng vuông góc với da để loại bỏ độ rời
LGI	Sử dụng các nhóm bất biến để loại bỏ chuyển động

hơn so với các phương pháp còn lại theo [6].

Phần trên đã khảo sát các thuật toán trích xuất rPPG sử dụng các phương pháp truyền thống. Các phương pháp có những ưu nhược điểm khác nhau. Theo các khảo sát, phương pháp sử dụng LGI và POS được cho là có độ chính xác cao nhất. Do đó, đồ án sẽ tập trung vào 2 phương pháp này. Như đã trình bày ở phần trước, tín hiệu rất nhạy bởi nhiều và bị ảnh hưởng nhiều bởi độ dày của bì mặt da, các chuyển động,... Vì vậy, việc chọn vùng thích hợp sẽ đảm bảo độ chính xác của tín hiệu. Đó là lý do chương này sẽ trình bày về kĩ thuật nhận dạng khuôn mặt và các vùng thích hợp để hạn chế những khuyết điểm trên.

4.4 Lựa chọn vùng ROI cho rPPG

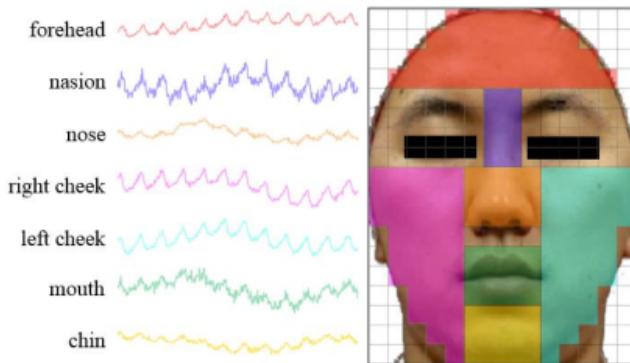
Việc lựa chọn các vùng Region of Interest tùy thuộc vào từng thuật toán. Theo [8], ta có bảng tóm tắt các vùng ROI cho từng thuật toán như sau:

Bảng 4.2: Các vùng ROI ứng với các phương pháp khác nhau

Thuật toán	GREEN	PCA / ICA	CHROM	POS	LGI
Vùng ROI	Khuôn mặt	Khuôn mặt	Khuôn mặt + da	Da	Da

Việc chọn ROI thích hợp sẽ giúp cho kết quả đo được không bị sai lệch bởi các yếu tố khác. Tùy thuộc vào các tính chất của thuật toán mà các phương pháp chọn ROI sẽ khác nhau. Thông thường, các vùng ROI được chọn sẽ là các vùng da (skin) do dễ dàng theo dõi được sự thay đổi của lưu lượng máu do tim gây ra hơn các vùng khác. Ngoài ra, quá trình đo lượng rPPG, có thể bị ảnh hưởng nhiều bởi chuyển động của cơ mặt, độ dày của da. Do đó, các vùng được chọn nên ít có sự thay đổi trong suốt quá trình đo.

Theo [10], đã khảo sát tỉ số công suất trên nhiều (Signal-to-noise ratio (SNR)) của các vùng khác nhau như Hình 4.4. Kết quả cho thấy vùng mũi (nose), vùng má trái (left cheek), vùng má phải (right cheek) cho kết quả SNR cao hơn các vùng còn lại: các vùng này gần như ít bị ảnh hưởng bởi các chuyển động của các cơ.



Hình 4.6: Khảo sát các vùng ROI

4.5 Kết luận chương

Chương này đã khảo sát cơ sở lý thuyết sinh học về kỹ thuật rPPG, cũng như khảo sát các thuật toán khác nhau GREEN, CHROM, POS, LGI,... Các phương pháp sẽ có những ưu và nhược điểm khác nhau. Nhưng nhìn chung ta có nhận xét rằng: GREEN dễ bị ảnh hưởng bởi nhiễu, CHROM và POS tập trung giải quyết vấn đề màu da. Trong khi đó, LGI tập trung giải quyết vấn đề nhiễu do chuyển động. Do đó, việc lựa chọn phương pháp nào để thực hiện cho mô hình dự đoán sẽ cần được khảo sát ở các chương sau. Chương tiếp theo sẽ thực hiện khảo sát về cơ sở lý thuyết các mô hình máy học. Từ đó, tiến hành xây dựng mô hình dự đoán cho bài toán phân loại video thật/ giả.

Chương 5

MỘT SỐ MÔ HÌNH MÁY HỌC VÀ CÁC PHƯƠNG PHÁP TỐI ƯU & ĐÁNH GIÁ MÔ HÌNH

Các chương trên đã giúp ta tìm được đặc trưng để huấn luyện với các mô hình. Chương này sẽ tìm hiểu các phương pháp máy học để sử dụng các dữ liệu từ tín hiệu rPPG làm dữ liệu đầu vào cho quá trình huấn luyện và kiểm thử. Ba phương pháp chính sẽ được khảo sát bao gồm: Support Vector Machine, Convolutional Neural Network, Recurrent Neural Network.

5.1 Một số mô hình máy học

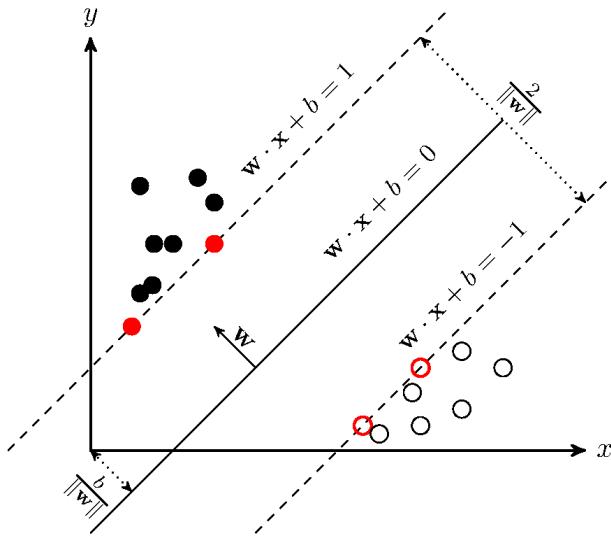
5.1.1 Support Vector Machine

Xét bài toán phân lớp (gồm hai lớp): nhiệm vụ của bài toán là tìm ra một ngưỡng Thres. Khi này nếu $y \leq \text{Thres}$, dữ liệu sẽ thuộc lớp A và ngược lại dữ liệu sẽ thuộc lớp B. Dựa trên ý tưởng đó, xét một siêu mặt phẳng có n chiều. Nhiệm vụ khi này là tìm một siêu mặt phẳng để phân lớp cho các nhóm dữ liệu. Hình 5.1 bên dưới sẽ minh họa cho cách hoạt động của thuật toán này.

Xét một mặt phẳng phân loại có dạng $y = w^T \mathbf{x} + b$ với w , b là các thông số của mặt phẳng. Ta có một điểm dữ liệu (\mathbf{x}_n, y_n) , để tính khoảng cách từ 1 điểm dữ liệu đến siêu mặt phẳng, ta có biểu thức như sau:

$$\frac{y_n |w^T \mathbf{x}_n + b|}{\|w\|_2} \quad (5.1)$$

Lưu ý rằng, ký hiệu \mathbf{x} đại diện cho vector $\mathbf{x} = [x_0, x_1, \dots, x_n]$



Hình 5.1: Minh họa cho thuật toán SVM

Margin trong SVM được định nghĩa là khoảng cách giữa siêu mặt phẳng đến 2 điểm dữ liệu gần nhất tương ứng với phân lớp.

$$Margin = \min_n \frac{y_n |w^T \mathbf{x}_n + b|}{\|w\|_2} \quad (5.2)$$

Nhiệm vụ của bài toán SVM: tìm mặt phẳng phân chia được 2 lớp dữ liệu thỏa mãn khoảng cách giữa hai điểm dữ liệu đến mặt phẳng là bằng nhau và Margin đạt giá trị lớn nhất.

$$(w, b) = \arg \max_{w, b} Margin \quad (5.3)$$

Ta quy ước rằng khoảng cách từ điểm gần nhất như sau:

$$\frac{y_n |w^T \mathbf{x}_n + b|}{\|w\|_2} = \frac{1}{\|w\|_2} \quad (5.4)$$

Từ đó, khoảng cách từ hai điểm thuộc hai lớp đến mặt phân lớp gần nhất (như Hình 5.1) là:

$$\frac{2}{\|w\|_2} \quad (5.5)$$

Khi này bài toán trở thành:

$$(w, b) = \arg \max_{w,b} \frac{2}{\|w\|_2} \quad (5.6)$$

$$\rightarrow (w, b) = \arg \min_{w,b} \frac{1}{2} \|w\|_2 \quad (5.7)$$

$$\rightarrow (w, b) = \arg \min_{w,b} \frac{1}{2} w^T w \quad (5.8)$$

sao cho $y_n |w^T \mathbf{x}_n + b| \geq 1$

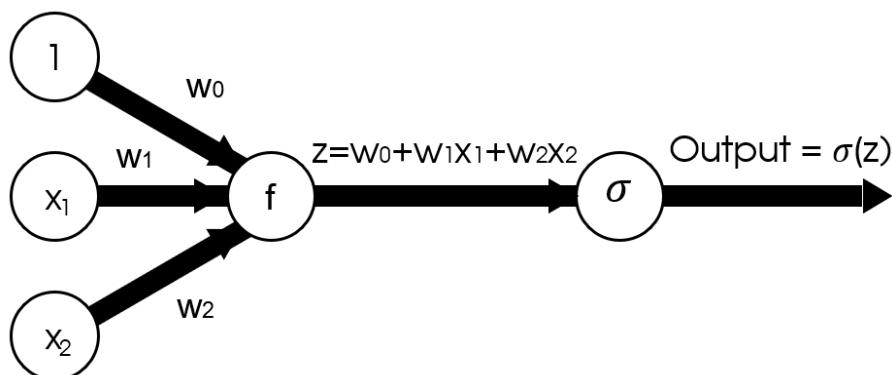
Để xác định lớp của một điểm dữ liệu ta xác định xem phía của dữ liệu so với mặt phẳng phân lớp. Nói cách khác, dựa trên dấu của biểu thức sau:

$$class(\mathbf{x}) = sgn(w^T \mathbf{x} + b) \quad (5.9)$$

Như vậy, thuật toán SVM là một trong những thuật toán cơ bản nhất được sử dụng để phân loại dữ liệu. SVM dựa trên một mặt phẳng để phân loại dữ liệu, thông thường SVM hiệu quả nhất đối với các bài toán phân loại 2 lớp. Đối với các bài toán phức tạp hơn, ta sẽ xem xét của dụng các phương pháp học sâu khác như ANN, CNN, RNN,...

5.1.2 Artificial Neural Network

Mạng Artificial Neural Network (ANN) hay còn được gọi tắt là Neural Network là một mô hình máy học, bao gồm nhiều layer, trong mỗi layer sẽ gồm các unit khác nhau. Dữ liệu đầu ra của lớp trước sẽ là dữ liệu đầu vào của lớp sau. Mỗi unit trong các class được đặc trưng bởi các trọng số (weight, bias) và hàm kích hoạt (một hàm phi tuyến như: sigmoid, ReLU, tanh, softmax,...). Các trọng số ban đầu được khởi tạo một cách ngẫu nhiên. Một mạng nơ-ron đơn giản được mô tả ở hình dưới.



Hình 5.2: Mạng ANN đơn giản

Theo Hình 5.2, quá trình hoạt động sẽ được mô tả đơn giản như sau:

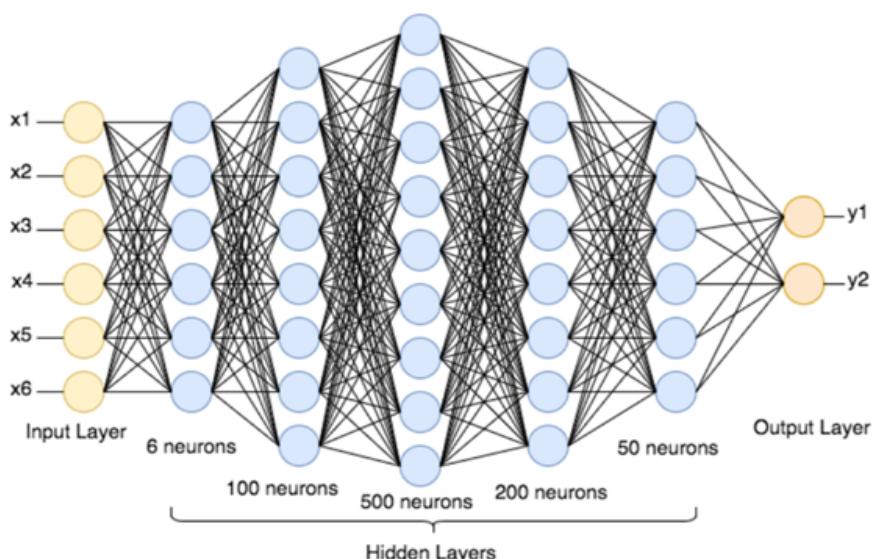
- Dữ liệu đầu vào sẽ được tổ hợp tuyến tính với các trọng số đã được thiết lập trước. Biểu thức như sau:

$$z = w_0 + w_1.x_1 + w_2.x_2 \quad (5.10)$$

- Sau đó, kết quả của Biểu thức (5.10) sẽ được đưa qua hàm kích hoạt (sigmoid, reLU, tanh,...) có biểu thức như sau:

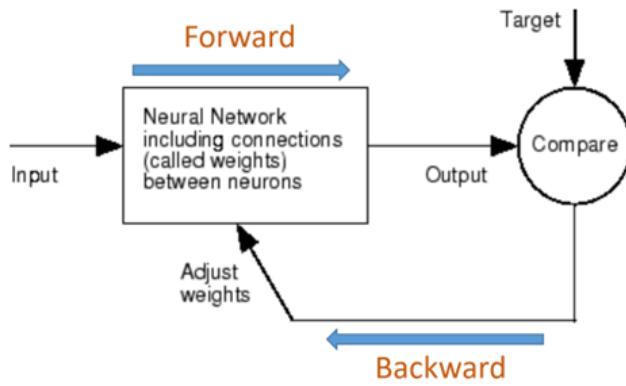
$$\hat{y} = \sigma(z) \quad (5.11)$$

Trên thực tế, một mạng Nơ-ron gồm rất nhiều class (Multi-Layer Neural Network) nhưng nhìn chung nguyên lý hoạt động vẫn như trên.



Hình 5.3: Mô hình Multi-Layer Neural Network

Quá trình huấn luyện mô hình gồm 2 quá trình chính: lan truyền thuận và lan truyền ngược. Quá trình trên thực hiện từ trái qua phải được gọi là quá trình lan truyền thuận. Quá trình này được sử dụng để tính toán kết quả ngõ ra. Tuy nhiên, độ chính xác của ngõ ra phụ thuộc rất nhiều vào trọng số. Do đó, việc huấn luyện nhằm mục đích để điều chỉnh trọng số đó. Quá trình này được gọi là lan truyền ngược (back propagation). Hình (5.4) sẽ mô tả cho quá trình này.



Hình 5.4: Quá trình lan truyền thuận & nghịch

Theo Hình 5.4, quá trình lan truyền ngược được thực hiện dựa trên 1 hàm mục tiêu nào đó và sử dụng các thuật toán tối ưu như Gradient Descent, Adam,...(phần này sẽ được trình bày bên dưới) để thỏa điều kiện của hàm mục tiêu (tối đa/ tối thiểu).

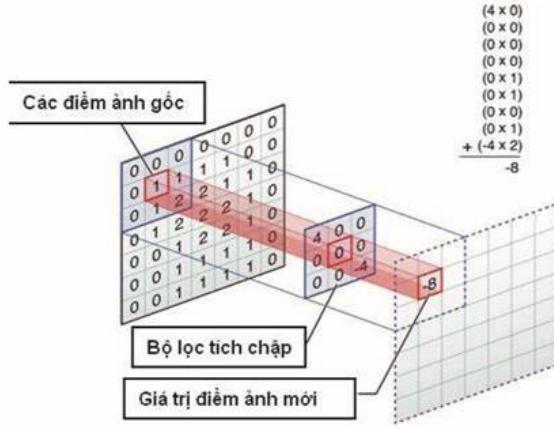
Tuy nhiên, điểm hạn chế lớn nhất của ANN là khối lượng tính toán sẽ rất lớn đặc biệt với ngõ vào là ảnh. Do đó, người ta để xuất CNN để giảm khối lượng tính toán, sau đó, sử dụng ANN cho việc huấn luyện mô hình. Phần bên dưới sẽ tiếp tục trình bày về CNN.

5.1.3 Convolutional Neural Network

Khi đầu vào là ảnh/ video hay dữ liệu dạng chuỗi, mạng ANN trở nên không còn phù hợp để tính toán do khối lượng thực hiện lớn. Do đó, ta cần sử dụng một phương pháp khác để trích xuất đặc trưng. Sau đó, dữ liệu mới được đưa vào mạng ANN. Trong cấu trúc mạng học sâu, có hai loại mô hình chính đó là Convolutional Neural Network (CNN) và Recurrent neural network (RNN). Mạng CNN thường được sử dụng phổ biến trong bài toán phân loại ảnh, xử lý video,... các dữ liệu đầu vào dưới dạng các ma trận. Trong khi đó, mạng RNN phù hợp cho bài toán có dạng chuỗi được ứng dụng trong lĩnh vực xử lý ngôn ngữ tự nhiên và xử lý tín hiệu,... Mạng Convolutional Neural Network sẽ trích xuất đặc trưng của ảnh thông qua các phép toán tích chập. Trong xử lý số tín hiệu, phép tính tích chập được tính bằng cách nhân tín hiệu $f(m)$ với nghịch đảo của $g(m)$. Tuy nhiên, trong CNN, việc tính tích chập đơn giản hơn nhiều. Ta lấy từng phần tử của kernel với các phần tử tương ứng trong ảnh (element-wise), sau đó thực hiện cộng các phần tử đó lại với nhau. Hình 5.5 mô tả quá trình tính tích chập.

Trên Hình (5.5), một cửa sổ có kích thước phù hợp (trong trường hợp này là 3x3) sẽ được áp lên từng vùng ảnh, từ đó, tính toán trích xuất ra những đặc trưng của ảnh. Cách tính tích chập được tính theo kiểu element-wise. Tương tự với các phương pháp học sâu khác, các lớp CNN cũng được cập nhật thông số thông qua quá trình lan truyền ngược. Ngoài ra, trong mạng CNN để giảm khối lượng tính toán, giữ lại các đặc trưng quan trọng người ta cũng thường dùng để Pooling.

Pooling sẽ được dùng để tính trung bình/ giá trị lớn nhất cho vùng ảnh đó. Giá trị trung



Hình 5.5: Quá trình tính tích chập trong CNN

bình/lớn nhất mang tính đại diện cho vùng ảnh đó. Như đã trình bày ở trên, mạng CNN thường được sử dụng chung với mạng ANN. Do đó, kết thúc mạng CNN, ta phải thực hiện việc duỗi thẳng (Flatten) ma trận để phù hợp kích thước ngõ vào mạng ANN (ngõ vào là vector)

Như vậy, mạng CNN được sử dụng phổ biến với bài toán với ngõ vào là hình ảnh. Ưu điểm của phương pháp này sẽ giúp giảm khối lượng tính toán của ANN do trích xuất được những đặc trưng quan trọng. Phần bên dưới sẽ trình bày mô hình còn lại trong máy học: Recurrent Neural Network

5.1.4 Recurrent Neural Network

Như đã trình bày ở trên, mạng Recurrent Neural Network phù hợp với dữ liệu dạng chuỗi và được ứng dụng trong xử lý tín hiệu. Đây là cấu trúc mạng được ứng dụng trong đồ án này. Phân bên dưới sẽ trình bày về tổng quan về RNN và mạng LSTM.

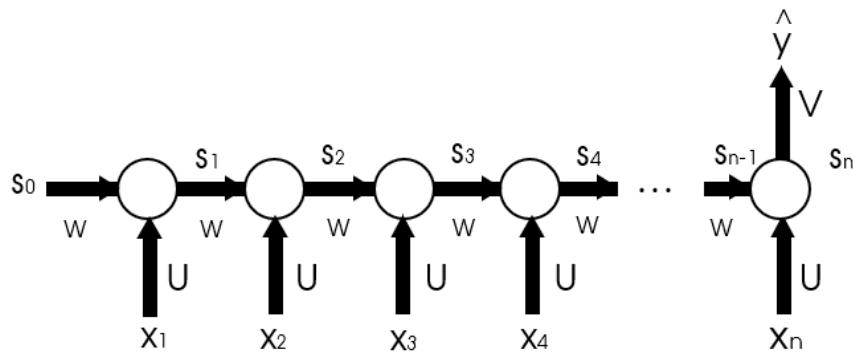
Tổng quan về RNN

Mạng RNN hoạt động dựa trên cơ chế xử lý từng tín hiệu đầu vào theo một thứ tự nhất định. Đây là bài toán thuộc loại many-to-one và many-to-many. Mô hình của RNN được mô tả như Hình 5.6:

Như trên Hình 5.6, do dữ liệu có dạng chuỗi nên ứng với từng ngõ vào x_1, x_2, \dots, x_n sẽ từng ứng với từng thời điểm của chuỗi. s đại diện cho các trạng thái (state), x đại diện cho các dữ liệu đầu vào. Khi chưa có dữ liệu đầu vào, ta sẽ có trạng thái s_0

Một trạng thái s được tính bởi biểu thức:

$$s_t = f(U \cdot x_t + W \cdot s_{t-1}) \quad (5.12)$$



Hình 5.6: Cấu trúc mạng RNN

Với f là hàm kích hoạt (activation function). Ngõ ra được tính theo biểu thức.

$$\hat{y} = g(V \cdot s_n) \quad (5.13)$$

Với g là hàm kích hoạt phi tuyến như: tanh, sigmoid,.. Quá trình huấn luyện mạng RNN: từ Hình 5.6, ta nhận thấy có 3 tham số cần tìm là W , U , V . Tương tự với mạng CNN, ta cũng có thể huấn luyện bằng thuật toán Gradient Descent theo thứ tự từ lớp cuối lên lớp đầu tiên (lan truyền ngược). Cụ thể như sau:

$$\frac{\partial L}{\partial V} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial V} \quad (5.14)$$

$$\frac{\partial L}{\partial W} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial s_n} \cdot \frac{\partial s_n}{\partial W} \quad (5.15)$$

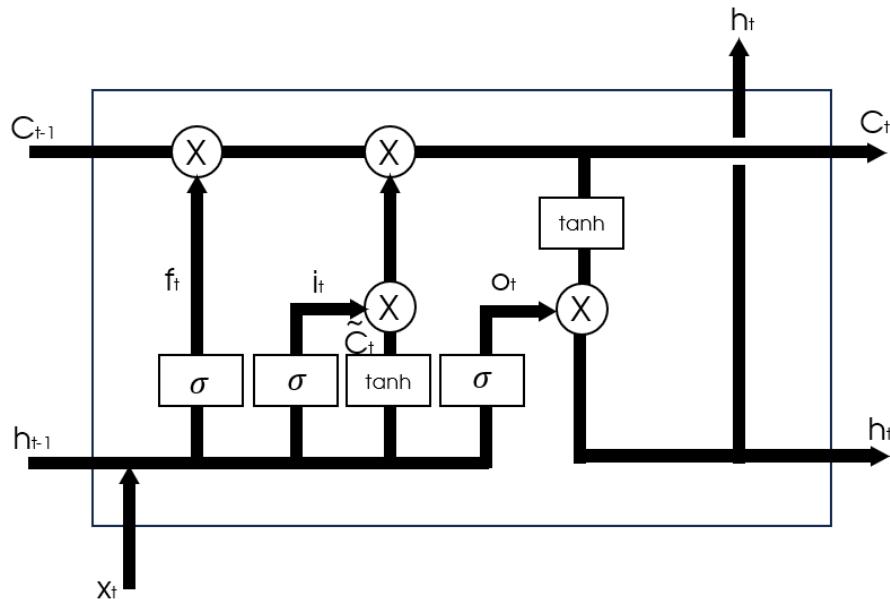
$$\frac{\partial L}{\partial U} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial x_n} \cdot \frac{\partial x_n}{\partial U} \quad (5.16)$$

Dựa vào biểu thức đạo hàm của W , U có thể thấy rằng biểu thức dễ xảy hiện tượng vanishing gradient do các lớp phía sau phụ thuộc vào các lớp trước. Hiện tượng vanishing gradient là hiện tượng khi mẫu số gần bằng 0 và giá trị đạt giá trị vô cùng lớn dẫn đến không học được hệ số mới.

Để khắc phục hạn chế này, người ta đã đưa ra mô hình Long short term memory (LSTM) hay Gated Recurrent Unit (GRU)

Long short term memory (LSTM)

Mô hình LSTM được minh họa bởi hình bên dưới:



Hình 5.7: Mô hình LSTM

Trong Hình 5.7: x_t là các ngõ vào, c_t là các cell state, h_t là các hidden state. σ, \tanh là các hàm kích hoạt. Quá trình thuận trên được tính toán như sau:

$$f_t = \sigma(x_t \cdot U_f + h_{t-1} \cdot W_f + b_f) \quad (5.17)$$

$$o_t = \sigma(x_t \cdot U_o + h_{t-1} \cdot W_o + b_o) \quad (5.18)$$

$$i_t = \sigma(x_t \cdot U_i + h_{t-1} \cdot W_i + b_i) \quad (5.19)$$

$$\tilde{c}_t = \tanh(x_t \cdot U_c + h_{t-1} \cdot W_c + b_c) \quad (5.20)$$

Với W_f, U_f là các hệ số weight cần tìm, b_f, b_i, b_o, b_c lần lượt là các hệ số bias. Từ các biểu thức (5.17), (5.18), (5.19), (5.20), ta có được ngõ ra cuối cùng như sau:

$$c_t = c_{t-1} \cdot f_t + \tilde{c}_t \cdot i_t \quad (5.21)$$

$$h_t = \tanh(c_t) \cdot o_t \quad (5.22)$$

Như đã trình bày ở phần trên, thuật toán LSTM giải quyết vấn đề vanish gradient. Dưới đây, sẽ mô tả chi tiết cách thuật toán. Ở thuật toán RNN, vấn đề vanish gradient do thành phần (5.15), (5.16) gây ra. Trên thực tế, biểu thức LSTM cũng gặp vấn đề vanish gradient tương tự RNN do $\frac{\partial(c_t)}{\partial(c_{t-1})} = f_t$. Đối với RNN, giá trị đạo hàm sẽ nằm trong [0,1]: khi nhiều giá trị nhỏ hơn 0 nhân với nhau sẽ tạo ra các giá trị vô cùng nhỏ gây ra hiện tượng vanish gradient. Tuy nhiên, do $f_t \approx 1$ nên hiện tượng vanish gradient ít xảy ra hơn.

Một cách tổng quát, có 3 phương pháp được dùng để phân loại: SVM, CNN và LSTM. SVM thường phù hợp cho bài toán phân loại hai lớp. CNN phù hợp với bài toán là các hình ảnh, tensor. Trong khi đó, LSTM phù hợp cho các bài toán có dạng chuỗi, tín hiệu.

Chương tiếp theo sẽ khảo sát các thuật toán được sử dụng để tối ưu các mô hình trên.

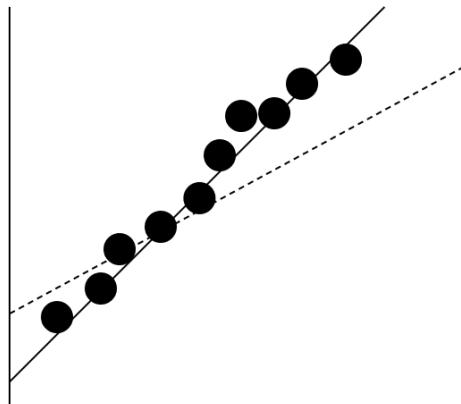
5.2 Thuật toán tối ưu

5.2.1 Gradient Descent

Xét một mô hình hồi quy tuyến tính đơn giản, được tính bởi biểu thức sau:

$$y = wx + b \quad (5.23)$$

Với x là một đại lượng ngõ vào, y là giá trị ngõ ra. w , b lần lượt là các trọng số của mô hình. Với mô hình hồi quy tuyến tính, ban đầu, các trọng số được khởi tạo ngẫu nhiên. Sau đó, với các tập dữ liệu có sẵn (x,y) mô hình sẽ cập nhật lại hệ số w,b sao cho phù hợp với dữ liệu đang có. Hình trên cho thấy đường nét đứt đại diện cho đường thẳng khởi tạo ban đầu, không phù



Hình 5.8: Mô hình hồi quy tuyến tính

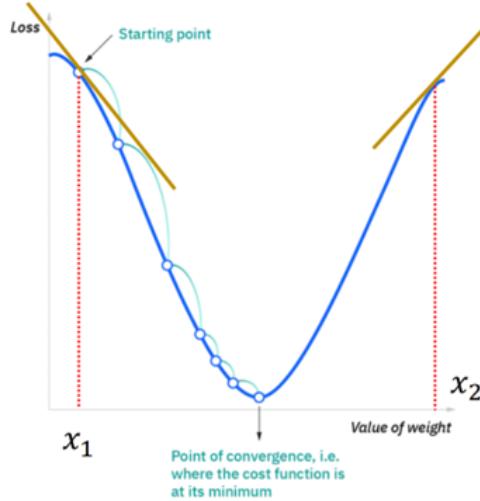
hợp với các điểm dữ liệu (các chấm đen). Đường thẳng đặc trưng bởi hệ số góc và bias, sau quá trình học các hệ số này được cập nhật tạo thành đường thẳng màu đen: phù hợp với tập dữ liệu.

Việc học và cập nhật các trọng số có nhiều phương pháp khác nhau: phổ biến nhất là Gradient Descent. Nguyên tắc hoạt động của phương pháp này được thực hiện dựa trên việc tính đạo hàm của hàm lỗi theo các trọng số (weight, bias). Sau đó, điều chỉnh trọng số theo hướng của gradient để hàm lỗi hội tụ tại điểm cực tiểu, được thể hiện ở biểu thức sau:

$$w \leftarrow w - \alpha \cdot \frac{\partial L}{\partial w} \quad (5.24)$$

Trong công thức trên, α là tỉ lệ học (learning rate) đóng vai trò như tốc độ học của thuật toán,

ký hiệu ∂ thể hiện cho đạo hàm. w đại diện cho các trọng số của mô hình. Ảnh bên dưới minh họa cho thuật toán: Như Hình trên, các trọng số được cập nhật sao cho hàm lỗi đạt giá trị cực

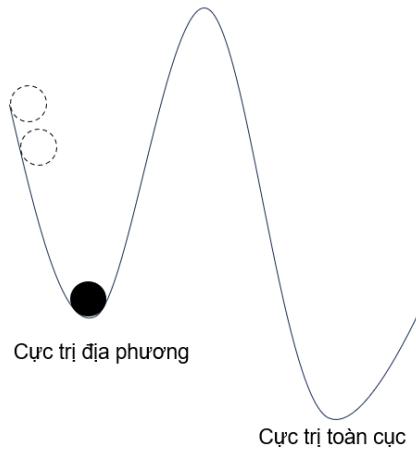


Hình 5.9: Thuật toán Gradient Descent

tiểu: khi này, ngõ ra dự đoán sẽ ít sai số hơn so với bộ dữ liệu huấn luyện.

5.2.2 Gradient Descent with Momentum

Hạn chế của phương pháp Gradient Descent là tốc độ hội tụ chậm và dễ dàng rơi vào điểm cực tiểu địa phương (local minimum), trong khi đó, ta mong muốn tìm được điểm cực tiểu toàn cục (global minimum). Minh họa ở hình bên dưới: Để giải quyết vấn đề đó, ta thêm vào biểu



Hình 5.10: Hạn chế của Gradient Descent

thức của Gradient Descent hệ số quan tính γ . Khi này, biểu thức trở thành:

$$v = \gamma v + \alpha \frac{\partial(w)}{\partial(w)} \quad (5.25)$$

$$w \leftarrow w - v \quad (5.26)$$

Ban đầu, ta khởi tạo ma trận quán tính $\Delta w = 0$. γ , α lần lượt là hệ số quán tính và hệ số học.

5.2.3 RMS prop

Việc điều chỉnh hệ số học đóng một vai trò quan trọng trong việc huấn luyện mô hình. Nếu hệ số quá nhỏ, tốc độ hội tụ sẽ rất chậm, ngược lại, nếu hệ số quá lớn có thể dẫn đến vấn đề quá khớp (overfitting). Do đó, Adagrad đã xem hệ số học như một tham số được cập nhật trong quá trình huấn luyện. Tuy nhiên Adagrad có một vài điểm hạn chế và được cải thiện bổ sung với thuật toán RMS prop. Thuật toán có biểu thức như sau:

$$E[g^2]_t = 0.9E[g^2]_{t-1} + 0.1g_t^2 \quad (5.27)$$

$$w \leftarrow w - \frac{\eta}{\sqrt{G_t + \epsilon}} g_t \quad (5.28)$$

Về bản chất đây là một dạng của Gradient Descent với hệ số học được cập nhật tự động. Do đó, nó vẫn mang hạn chế rằng dễ dàng rơi vào cực trị địa phương so với phương pháp momentum.

5.2.4 Adam

Thuật toán Adam là một sự kết hợp của thuật toán Momentum và RMSprop: nó sử dụng 2 momentum trong việc huấn luyện như biểu thức sau:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (5.29)$$

$$v_t = \beta_2 m_{t-1} + (1 - \beta_2) g_t^2 \quad (5.30)$$

Trong biểu thức (5.29) đóng vai trò trong việc cập nhật hệ số học, (5.30) đóng vai trò như momentum tránh các cực trị địa phương. Như vậy biểu thức cập nhật hệ số như sau:

$$w \leftarrow w - \alpha \frac{m_t}{\sqrt{v_t + \epsilon}} \quad (5.31)$$

Trong biểu thức (5.31), w là trọng số cần cập nhật, α là hệ số học, m_t và v_t được tính theo biểu thức (5.29) và (5.30), ϵ là một số vô cùng nhỏ để tránh hiện tượng chia cho 0 xảy ra.

Phần trên đã khái quát các thuật toán phổ biến. Trong đó, Adam được sử dụng phổ biến trong huấn luyện mô hình do ưu điểm có thể tránh được cực tiểu địa phương. Phần tiếp theo sẽ khảo sát phương pháp Deep supervision giúp cải thiện độ chính xác của mô hình.

5.3 Deep supervision

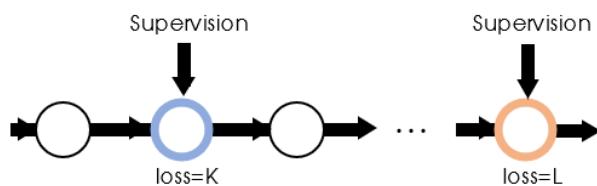
Deep Supervision là một kĩ thuật cải thiện khả năng học của mô hình học sâu. Thông thường, một mô hình sẽ chỉ đưa ra kết quả học ở lớp cuối cùng (tức các quá trình học tập diễn ra theo trình tự từ lớp đầu đến lớp cuối). Khác với phương pháp đó, Deep Supervision chia mô hình thành nhiều head buộc mô hình phải học đưa ra kết quả từ những lớp giữa. Điều này giúp mô hình buộc phải học nhiều hơn và chính các lớp giữa đã có thể đưa ra kết quả. Có thể xem đó là các bộ phân loại con của mô hình. Việc sử dụng Deep Supervision giúp mô hình cải thiện hai vấn đề chính:

- Độ chính xác của mô hình
- Làm "mượt" việc tính gradient, giải quyết vấn đề Vanish Gradient.

Nhìn chung có 3 dạng của Deep Supervision: Hidden Layer Deep Supervision, Different Branches Deep Supervision, Deep Supervision Post Encoding [11]

5.3.1 Hidden Layer Deep Supervision (HLDS)

Dạng này áp dụng với mạng neural network không rẽ nhánh. Các hidden layer được thực hiện lần lượt từ trái sang phải. Một số hidden layer được sử dụng kĩ thuật Deep Supervision để tăng độ chính xác mô hình. Được minh họa ở hình dưới: Như Hình 5.3.1, mạng neural sẽ có



Hình 5.11: Hidden Layer Deep Supervision

dạng nối tiếp và có một hoặc nhiều neuron được chọn để thực hiện Deep Supervision. Như vậy, khi này sẽ có $(n+1)$ hàm mục tiêu (hàm lỗi) được sử dụng để tối ưu mô hình, với n là số lượng neuron được dùng để tính Deep supervision. Khi này, hàm lỗi ở ngõ ra cuối cùng sẽ có dạng:

$$L = L_L + \alpha \cdot L_k \quad (5.32)$$

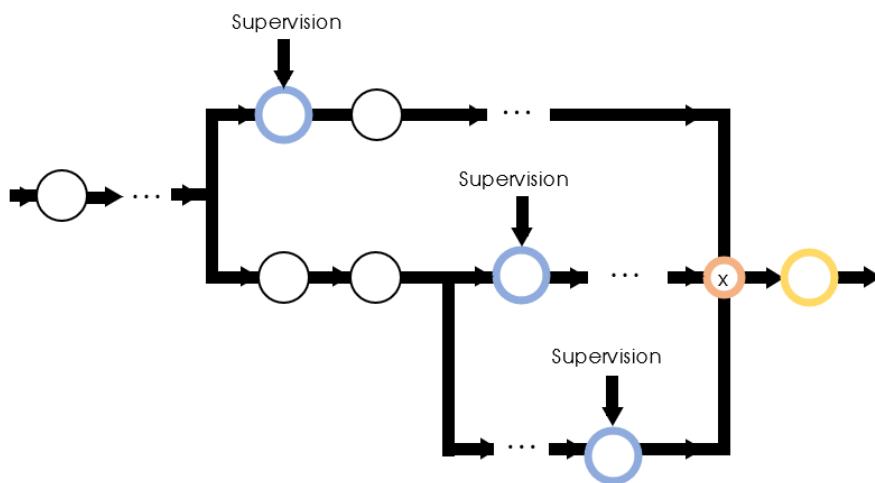
Trong Biểu thức (5.32), L là giá trị lỗi của toàn mô hình, L_L là giá trị hàm lỗi của ngõ ra cuối cùng, L_k là giá trị hàm lỗi của neuron sử dụng Deep supervision, α là hệ số điều chỉnh ảnh hưởng của Deep supervision. Từ đó, ta có biểu thức cập nhật hệ số weight và bias được tính như sau:

$$w = \begin{cases} w - \eta \cdot \frac{\partial L_L}{\partial w}, & l = K + 1, \dots, L \\ w - \eta \cdot \left(\frac{\partial L_L}{\partial w} + \frac{\partial L_K}{\partial w} \right), & l = 1, \dots, K \end{cases} \quad (5.33)$$

Trong Biểu thức (5.33), ta nhận thấy rằng do quá trình lan truyền ngược nên các layer ở phía sau Deep supervision (tính từ trái sang phải) không bị ảnh hưởng trong bởi L_K . Ngược lại, các layer trước đó, sẽ được điều chỉnh bởi giá trị L_L và L_K điều này giúp cải thiện tốc độ học của mô hình và giảm hiện tượng vanish gradient do sự thêm vào của L_K .

5.3.2 Different Branches Deep Supervision (DBDS)

Dạng này được áp dụng với mạng có cấu trúc rẽ nhánh, các nhánh được áp dụng Deep Supervision để học các đặc trưng khác nhau. Sau đó, được tổng hợp lại với mạng chính và đưa ra dự đoán. Deep Supervision được sử dụng trong trường hợp này thích hợp cho việc cải thiện hiệu quả hơn là giải quyết vấn đề vanish gradient. Hình dưới sẽ mô tả cho lý thuyết trên: Như



Hình 5.12: Different Branches Deep Supervision

vậy, biểu thức hàm lỗi sẽ được tính như sau:

$$L = L_L + \sum \alpha.(L_k) \quad (5.34)$$

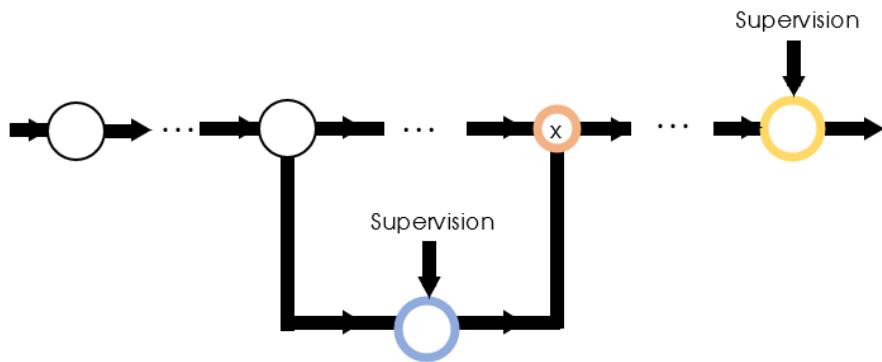
Với L_L , L_K , và L lần lượt là giá trị hàm lỗi ngõ ra layer cuối, giá trị hàm lỗi của Deep supervision, và giá trị hàm lỗi của toàn bộ mô hình. Từ đó, biểu thức cập nhật hệ số được tính như sau:

$$w = w - \eta \cdot \left(\frac{\partial L_L}{\partial w} + \frac{\partial L_K}{\partial w} \right) \quad (5.35)$$

5.3.3 Deep Supervision Post Encoding

Phương pháp này có điểm tương tự với phương pháp HLDS. Tuy nhiên, phương pháp này khác ở điểm, giá trị loss được tiếp tục sử dụng kết hợp với giá trị của layer trước đó qua một hàm số g nào đó. Đây là phương pháp kết hợp giữa phương pháp HLDS và phương pháp DBDS. Mô hình được thể hiện ở bên dưới:

Từ đây, ta xây dựng hàm lỗi với cách tính tương tự với hai mô hình trên. Ta được biểu thức



Hình 5.13: Deep Supervision Post Encoding

như sau:

$$L = L_L + \sum \alpha.(L_k) \quad (5.36)$$

Biểu thức cập nhật hệ số như sau:

$$w = \begin{cases} w - \eta * \frac{\partial L_L}{\partial w}, l = K + m, \dots L \\ w - \eta * (\frac{\partial L_L}{\partial w} + \frac{\partial L_K}{\partial w}), l = 1, \dots, K + m - 1 \end{cases} \quad (5.37)$$

Trong trường hợp này, giá trị $K+m$ chỉ vị trí giá trị nhánh của Deep supervision được nối với mô hình chính.

Tùy vào mô hình đang sử dụng mà ta sẽ quyết định sử dụng các hình thức khác nhau của kỹ thuật Deep supervision. Nhìn chung, kỹ thuật này hữu ích khi giải quyết được vấn đề vanish gradient cũng như cải thiện được độ chính xác của thuật toán.

Mỗi phương pháp đều có những ưu và nhược điểm riêng. Nhưng nhìn chung phương pháp sử dụng Deep supervision thực hiện tốt việc cải thiện hiệu suất mô hình và giảm hiện tượng vanish gradient.

5.4 Thông số đánh giá

5.4.1 Confusion matrix

Bài toán nhận dạng video Deepfake là bài toán phân loại hai lớp. Để đánh giá, người ta sử dụng ma trận nhầm lẫn (confusion matrix). Ma trận có dạng như sau:

Như trên Hình 5.14, minh họa cách thức mà ma trận nhầm lẫn đánh giá mô hình phân loại. Kết quả đánh giá dựa trên kết quả thực tế và kết quả dự đoán của mô hình. Các ô True Positive, True Negative, False Negative, False Positive chính là số lượng các dữ liệu thuộc về các loại trên. Để phân loại thành các loại trên, ta có các mô tả như sau:

		Thực tế	
		Positive	Negative
Dự đoán	Positive	True Positive	False Negative (Sai lầm loại II)
	Negative	False Negative (Sai lầm loại I)	True Negative

Hình 5.14: Ma trận nhầm lẫn

- True Positive, True Negative: Kết quả dự đoán cho ra giống với kết quả thực tế. Khi kết quả dự đoán và thực tế cùng là Positive, dữ liệu đó sẽ là một True Positive. Và ngược lại, dữ liệu đó sẽ là True Negative.
- False Negative: Kết quả thực tế là Negative trong khi đó mô hình dự đoán là Positive.
- False Positive: Ngược lại với False Negative, khi kết quả thực tế là Negative nhưng mô hình lại dự đoán là Positive.

Trong phân tích thống kê, thuật ngữ "Sai lầm loại I" và "Sai lầm loại II" được sử dụng trong việc kiểm định một giả thuyết. Trong ma trận này, False Positive chính là sai lầm loại I, trong khi đó, False Negative thuộc sai lầm loại II. Để chứng minh rằng, mô hình đạt yêu cầu (tức thỏa mãn một giả thuyết nào đó), các giá trị False Negative và False Positive đạt giá trị nhỏ càng tốt

5.4.2 Precision

Chỉ số Precision phản ánh số lần mô hình dự đoán là Positive đúng so với tổng số lần dự đoán Positive. Được đặc trưng bởi biểu thức sau:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (5.38)$$

Với biểu thức này nếu Precision = 1 khi này FP = 0, chứng tỏ mô hình có không có bất cứ dự đoán sai các Postive.

5.4.3 Recall

Recall còn được gọi là độ nhạy, được đặc trưng bởi biểu thức sau:

$$\text{Recall} = \frac{TP}{TP + FN} \quad (5.39)$$

Khi Recall = 1, tức FN = 0: khi này không có bất kỳ Negative nào bị dự đoán sai.

5.4.4 F1-Score

Mục tiêu của mô hình phân lớp phải tối ưu có Recall và Precision đạt kết quả càng gần giá trị 1 càng tốt. Tuy nhiên giữa hai giá trị này có mối tương quan ngược với nhau. Để hiểu được sự tương quan này, ta quay lại với ma trận nhầm lẫn ở phần trên.

- Precision: Khi Precision cao, khi này tỉ lệ False Positive giảm. Tương tự đối với tình huống ngược lại
- Recall: Khi Recall cao, khi này tỉ lệ False Negative giảm. Tương tự đối với tình huống ngược lại

Hai đại lượng trên không đánh giá được toàn diện mô hình. Do đó, đại lượng F1-Score được sử dụng để đặc trưng cho 2 giá trị trên và được xác định bởi biểu thức:

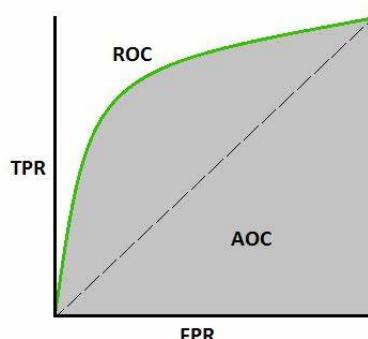
$$F1-score = \frac{2 * (precision * recall)}{(precision + recall)} \quad (5.40)$$

5.4.5 Area Under the Curve (AUC)

Trong bài toán phân lớp, ngoài việc sử dụng thông số độ chính xác (accuracy) để đánh giá khả năng phân lớp của mô hình, ta còn sử dụng AUC để đánh giá. Thông số sử dụng một đường cong thể hiện khả năng phân lớp theo các ngưỡng. Sau đó, tính diện tích phần bên dưới đường cong để đánh giá mô hình. Đường cong này gọi là ROC và được xây dựng dựa trên hai thông số: True positive rate (TPR) và False positive rate (FPR). Các thông số đó được tính theo công thức sau:

$$Rate = \frac{\text{type}}{\sum \text{predict}} \quad (5.41)$$

Với type là True positive hoặc là False positive, predict là tổng số lần dự đoán đúng hoặc tổng số lần dự đoán sai. Dựa vào hai thông số trên ta vẽ được đường cong và tính diện tích phần bên dưới của nó. Ta xem xét hình sau:



Hình 5.15: AUC và đường cong ROC

Dựa vào Hình 5.15, mô hình đạt TPR càng cao thì FPR càng thấp. Do đó, chỉ số AUC lý tưởng sẽ đạt giá trị 1.0, trong khi đó, mô hình không có tính phân loại (theo xác suất) sẽ đạt giá trị 0.5.

Việc đánh giá mô hình chỉ dựa trên thông số độ chính xác còn nhiều hạn chế và thiếu tính khách quan. Do đó, cần sử dụng nhiều thông số khác để đánh giá một cách toàn diện cho mô hình. Đối với bài toán phân lớp các thông số như AUC, Precision, F1-Score, Confusion matrix được sử dụng phổ biến.

5.5 Kết luận chương

Tóm lại, mô hình CNN thường được sử dụng với bài toán với đầu vào là ảnh hoặc ma trận, trong khi đó, RNN được sử dụng với bài toán có dữ liệu dạng chuỗi. Mạng ANN thông thường được sử dụng chung với CNN và RNN mà ít khi sử dụng đơn lẻ. Về phương pháp tối ưu mô hình, thuật toán Adam thường được sử dụng phổ biến do ưu điểm có thể tự điều chỉnh tốc độ học (learning rate) và tránh được cực tiểu địa phương của hàm lỗi giúp đạt được kết quả học tốt nhất. Tuy nhiên, trên thực tế, ta cần thực hiện khảo sát các phương pháp tối ưu khác nhau để tìm được phương pháp phù hợp nhất với mô hình. Để tăng tốc độ học của mô hình cũng như cải thiện độ chính xác của mô hình, đồ án cũng xem xét đến kĩ thuật Deep supervision. Đây là một kĩ thuật buộc mô hình phải đưa dự đoán từ những lớp giữa. Cuối cùng, chương đã khái quát các thông số thường được dùng để đánh giá bài toán phân loại

Chương 6

MÔ HÌNH LSTM SỬ DỤNG TÍN HIỆU RPPG

Dựa trên cơ sở lý thuyết trên, đồ án đề xuất một phương pháp để nhận dạng video Deepfake như sau. Quá trình gồm 3 bước chính: tạo lập bộ dữ liệu, xây dựng thuật toán, kiểm thử thuật toán và cải tiến.

6.1 Dữ liệu

Phương pháp sử dụng các thuật toán xử lý ảnh, xử lý số tín hiệu và máy học, do đó, cần một tập dữ liệu để huấn luyện mô hình và kiểm thử. Các video gốc được sử dụng trên tập dữ liệu Faceforensics++ và Celeb DF kết hợp với các video được thu thập thực tế. Các hình ảnh mặt người được thu thập từ nhiều nguồn trên Internet. Dựa vào dữ liệu đó, video giả được tạo ra bằng mô hình Thin-Plate Spline Motion Model for Image Animation. Video giả được huấn luyện tại website: <https://replicate.com/yoyo-nb/thin-plate-spline-motion-model>

Bộ dữ liệu gồm 1261 video được chia thành 3 tập dữ liệu: train, validation và test theo tỉ lệ 60:15:25. Cụ thể như sau:

Bảng 6.1: Cấu trúc dữ liệu

	Train	Validation	Test
Video thật	325	139	198
Video giả	427	183	193
Tổng	1208	322	391

6.2 Thuật toán đề xuất

6.2.1 Lựa chọn thuật toán trích xuất rPPG

Các phương pháp sử dụng mô hình học sâu cho độ chính xác cao với việc đo lường người thật, tuy nhiên, các phương pháp có những hạn chế như sau:

- Độ phức tạp của thuật toán lớn, yêu cầu phần cứng lớn.
- Theo [18], cho thấy rằng một số thuật toán học sâu có thể đưa ra dự đoán tín hiệu rPPG cho video không phải là người. Tức là không thể phân biệt được sự khác biệt giữa rPPG người thật và vật/ deepfake.

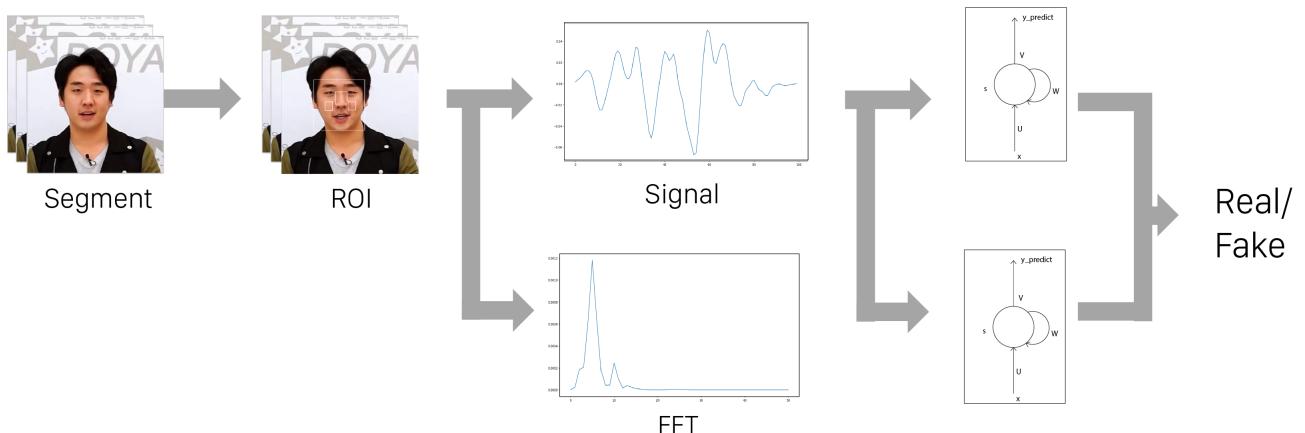
Do đó, đồ án chọn sử dụng các phương pháp xử lý ảnh truyền thống. Như đã phân tích ở phần 4.3 về các ưu và khuyết điểm của từng phương pháp. Do đó, đồ án thực hiện trên phương pháp CHROM, POS và LGI và so sánh hiệu quả giữa chúng.

Sau khi trích xuất tín hiệu rPPG, ta sử dụng các tín hiệu như là các đặc trưng để phân lớp.

6.2.2 Thiết kế mô hình

Mô hình LSTM

Sơ đồ giải thuật được thể hiện ở hình bên dưới:

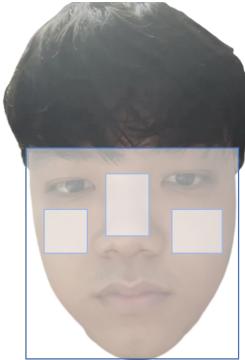


Hình 6.1: Mô hình sử dụng LSTM

Video đầu vào được chia thành n phân đoạn (segment). Ta sẽ làm việc trên các phân đoạn này, mỗi phân đoạn có độ dài $\omega=150$ (phần sau sẽ khảo sát mức độ ảnh hưởng của ω). Với các phân đoạn, ta thực hiện việc phát hiện khuôn mặt (face detection) và tìm (hai bên má và mũi). Cụ thể như sau:

Với video đầu vào, công cụ Dlib sẽ được sử dụng để nhận dạng mặt người và tìm ra các điểm trên khuôn mặt landmark.

Các vùng ROI sẽ được chọn như sau:



Hình 6.2: Vùng ROI

Các vùng má và mũi ít bị ảnh hưởng bởi chuyển động cơ mặt nên trong trường hợp này ta sử dụng chúng là vùng để thu tín hiệu.

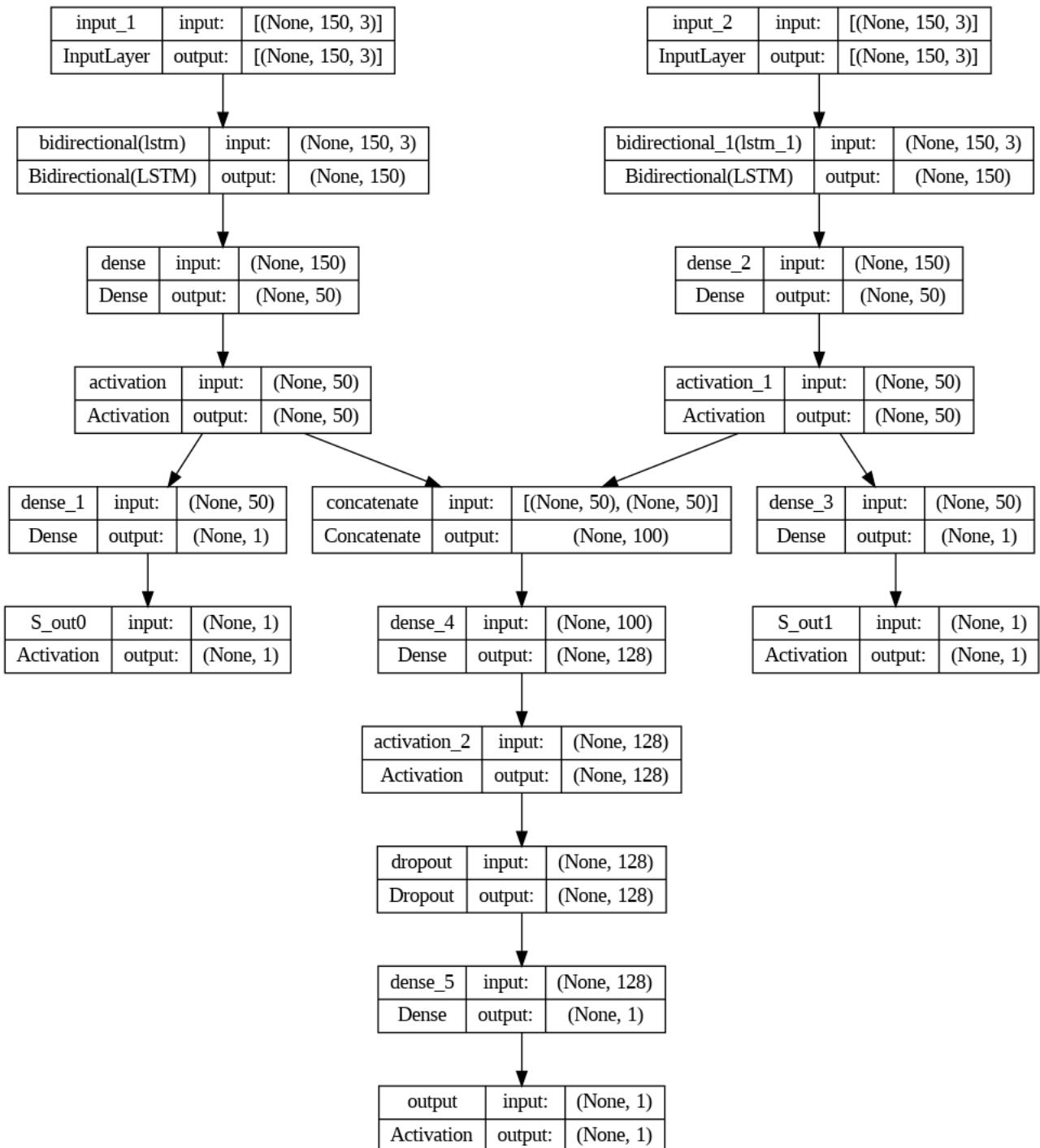
Các vùng ROI và cả khuôn mặt sẽ được dùng để trích xuất đặc trưng rPPG bằng các thuật toán (CHROM, LGI, POS). Theo nghiên cứu của [2], các thông tin trong miền tần số cũng mang các thông tin hữu ích cho việc phân loại. Do đó, các đặc trưng này cũng được sử dụng thông qua biến đổi Fourier. Từ hai đặc trưng trong miền không gian và tần số, ta đưa các đặc trưng này vào mạng LSTM kết hợp với kỹ thuật Deep Supervision để cải thiện hiện xuất. Mô hình chi tiết được thể hiện như Hình 6.3.

Như hình trên, tín hiệu trong miền thời gian và biến đổi Fourier trong miền tần số được đưa vào 2 nhánh riêng biệt của mạng và sử dụng Deep supervision để cải thiện hiệu suất mô hình. Sau khi đã được trích xuất đặc trưng, ta sẽ kết hợp hai nhánh này lại với nhau. Sau đó, sử dụng các neural network để phân lớp mô hình. Các hàm kích hoạt ở mô hình trên đều là Leaky ReLU. Lớp cuối cùng của mạng sẽ có hàm kích hoạt là sigmoid để biểu thị cho giá trị xác suất (giá trị từ 0 đến 1): mức 0 ứng với video thật và mức 1 ứng với video giả. Trên thực tế, các giá trị dự đoán sẽ nằm trong khoảng [0,1]. Do đó, ta dùng ngưỡng 0.5 để phân loại. Cụ thể:

$$\begin{cases} \hat{y} \geq 0.5 : fake \\ \hat{y} < 0.5 : real \end{cases} \quad (6.1)$$

Ngõ ra của mô hình là kết quả phân loại cho các phân đoạn. Để thu được kết quả của toàn bộ video, đồ án sẽ khảo sát bằng 2 phương pháp:

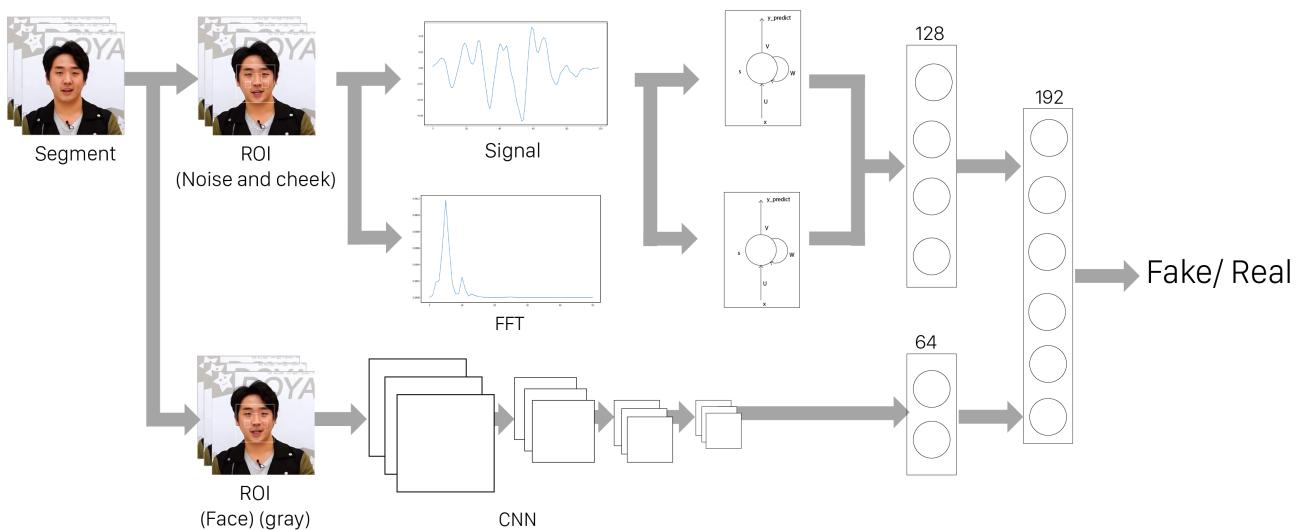
- Tính trung bình tất cả kết quả của các phân đoạn, sau đó, dùng ngưỡng để phân loại
- Kết quả của video được thực hiện theo nguyên tắc bỏ phiếu. Loại video sẽ được quyết định theo số lượng phân đoạn thật và giả.



Hình 6.3: Mô hình chi tiết

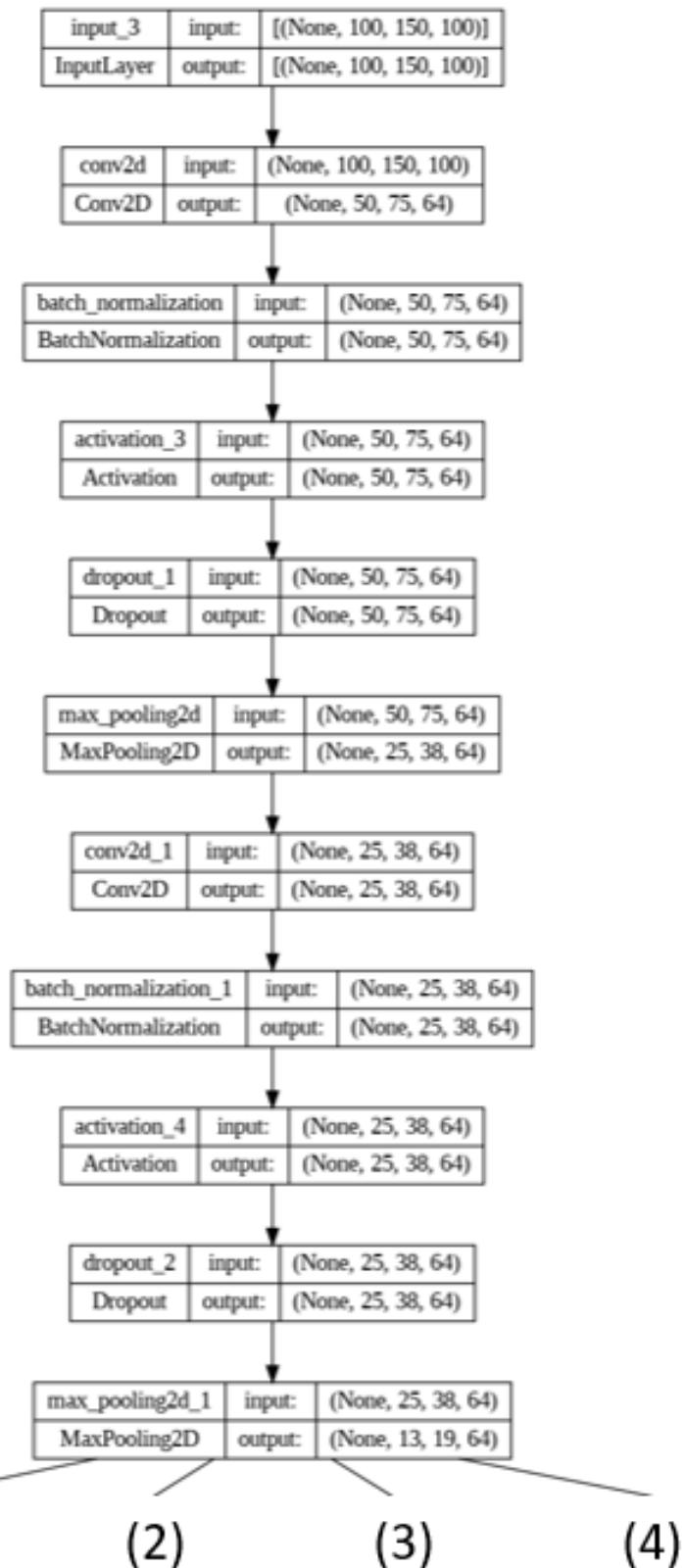
Mô hình LSTM kết hợp với CNN

Các đặc trưng bìe ngoài (khuôn mặt, tóc, miệng, các chuyển động) có thể mang lại những thông tin hữu ích cho mô hình phân loại video thật/ giả. Do đó, đồ án cũng xem xét việc bổ sung thêm đặc trưng để cải thiện độ chính xác. Dựa trên ý tưởng rằng các chuyển động trong các video, hình ảnh trong video giả mặc dù đã được cải thiện rất nhiều để giống với video thật tuy nhiên vẫn còn nhiều hạn chế. Đây được xem lại một yếu tố có thể khai thác để tăng độ chính xác của thuật toán. Mô hình được minh họa ở hình bên dưới: Các đặc trưng

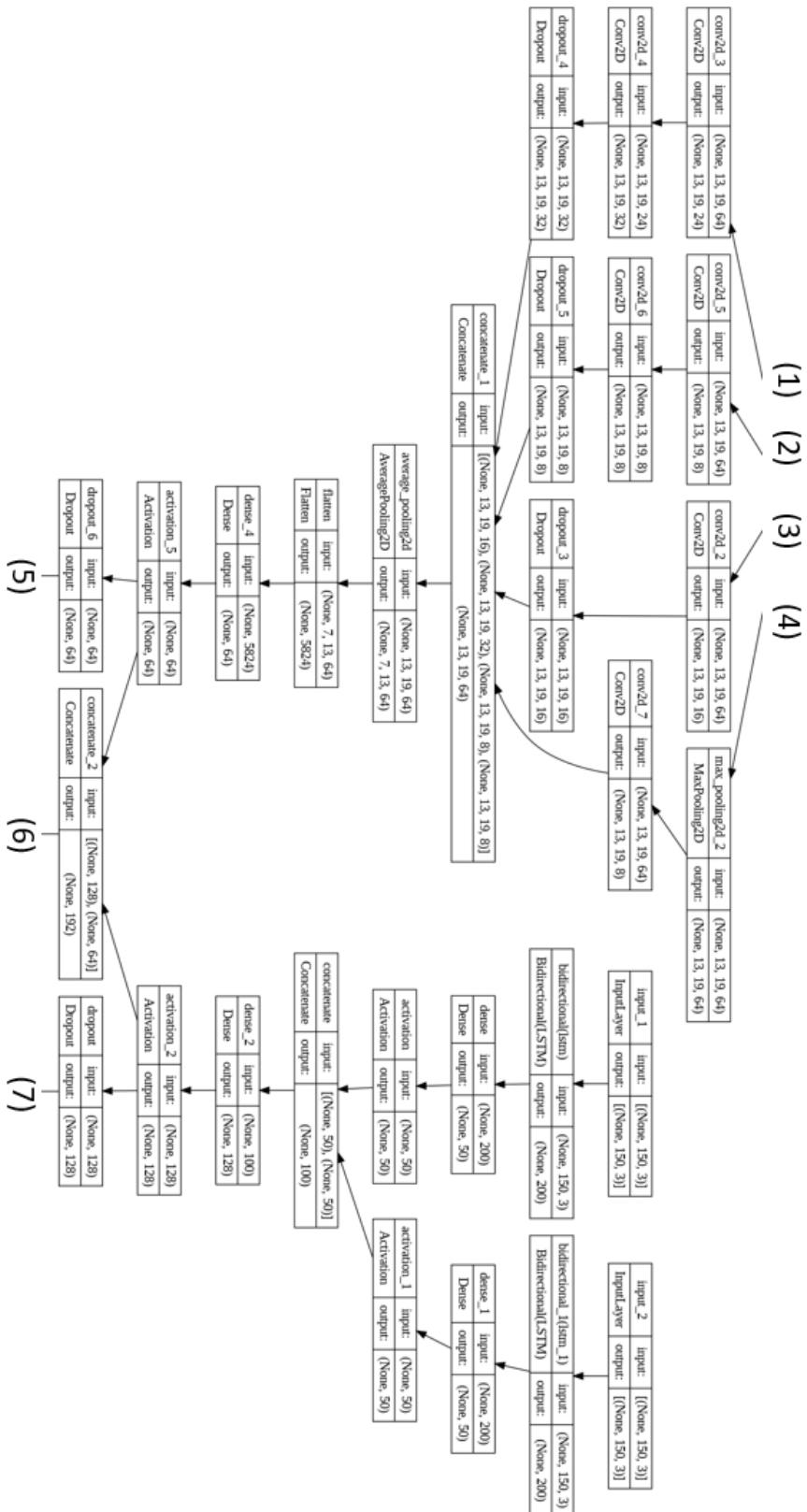


Hình 6.4: Mô hình cải tiến kết hợp với đặc trưng bìe ngoài

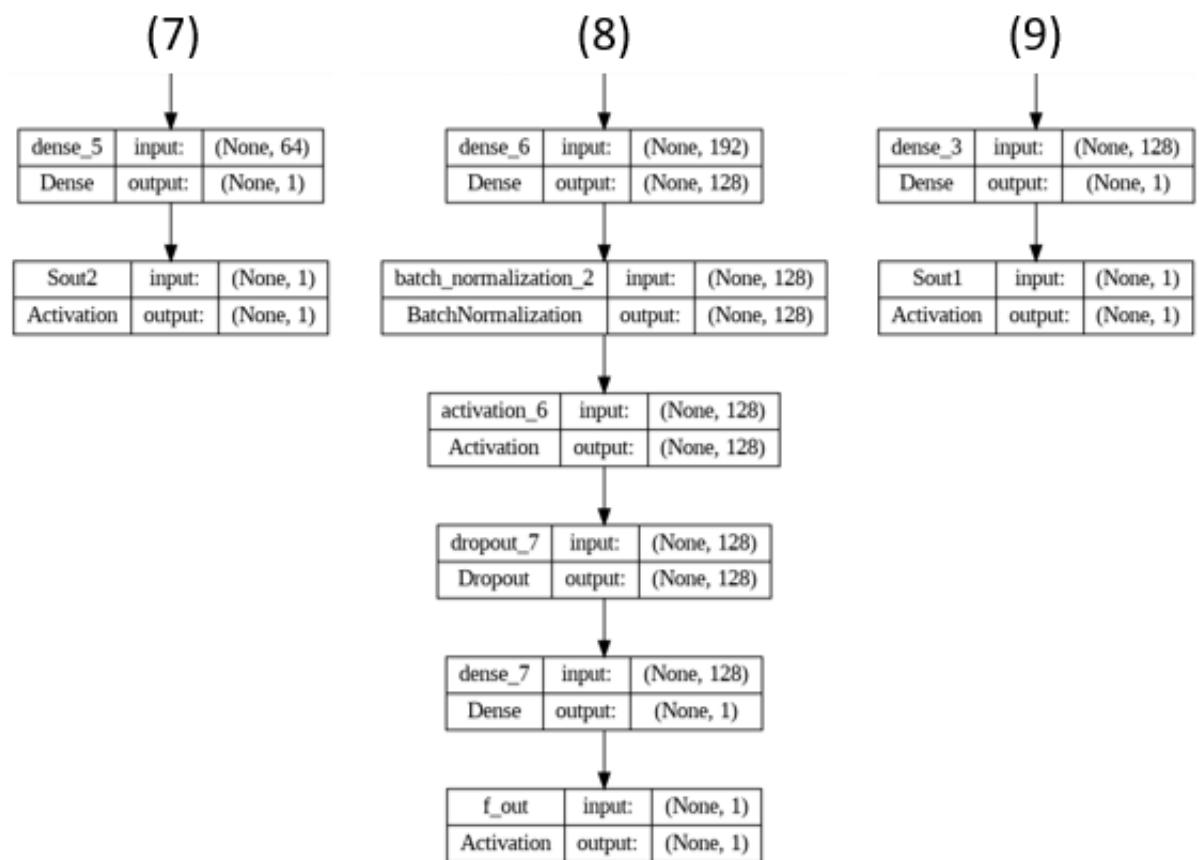
bìe ngoài được trích xuất với 100 khung hình chứa khuôn mặt. Để giảm khối lượng tính toán, các khung hình này được chuyển từ hệ màu RGB sang ảnh xám (gray). Kích thước khung hình được căn chỉnh với kích thước 150x100. Như vậy, ngõ vào của mạng mới này sẽ có kích thước (100,150,100) được đưa vào mạng InceptionV3 được tùy chỉnh số layer để giảm khối lượng tính toán. Như Hình 6.4) và , mô hình này sẽ được ra một vector đặc trưng có kích thước bằng (64,1). Vector đặc trưng này sẽ được kết hợp với vector đặc trưng có kích thước (128,1) của mạng LGI+LSTM bên trên tạo thành vector có kích thước (192,1). Sau đó, vector này sẽ qua lớp Dense với Dropout=0.5 để hạn chế vấn đề quá khớp (overfitting). Cuối cùng, ngõ ra sẽ qua hàm sigmoid để đưa ra kết quả phân loại 2 lớp. Mô hình mới này tiếp tục được sử dụng kĩ thuật Deep supervision để cải thiện tốc độ huấn luyện và độ chính xác. Mô hình cụ thể về các lớp sẽ được minh họa bên dưới:



Hình 6.5: Mô hình chi tiết kết hợp LSTM và CNN đoạn đầu



Hình 6.6: Mô hình chi tiết kết hợp LSTM và CNN đoạn giữa



Hình 6.7: Mô hình chi tiết kết hợp LSTM và CNN đoạn cuối

6.3 Kiểm thử thuật toán và cải tiến

Dồ án sử dụng ma trận nhầm lẫn, F1-score, Recall, Precision để đánh giá mô hình. Ngoài ra, đồ án so sánh thời gian xử lý, độ chính xác giữa các mô hình, đầu vào video khác nhau. Dồ án cũng thực hiện khảo sát các phương pháp trích xuất đặc trưng khác nhau, thử nghiệm trong các trường hợp khác nhau để tìm được mô hình tối ưu nhất.

6.4 Kết luận chương

Chương trên đã khái quát quá trình xây dựng bộ dữ liệu và huấn luyện và đánh giá mô hình. Một cách tổng quát, video đưa vào sẽ được chia thành các phân đoạn, sau đó, thực hiện việc phát hiện khuôn mặt và trích xuất ROI. Từ các ROI thực hiện trích xuất rPPG và cuối cùng đưa vào mạng LSTM để thực hiện việc phân loại.

Chương 7

KẾT QUẢ VÀ PHÂN TÍCH

7.1 Phương pháp tiếp cận

Các video đầu vào được trích xuất tín hiệu rPPG thông qua các vùng ROI. Sau đó, tín hiệu được xem là các đặc trưng thông qua thuật toán LSTM. Để đánh giá tính hiệu, đồ án sử dụng độ chính xác, ma trận nhầm lẫn, Precision, Recall, F1-score.

7.2 Thiết lập thông số

Mô hình được huấn luyện sử dụng các thông số được thiết lập như sau:

- **Hàm lỗi:** Binary Crossentropy
- **Thuật toán tối ưu:** Adam
- **Learning rate** 1e-4
- **Đánh giá:** Accuracy
- **Batch size:** 128
- **Tỉ lệ train-valid:** 7:3

Mô hình được thực hiện lấy mẫu trích xuất tín hiệu rPPG trên máy tính Intel (R) Core(TM) i5-8300H CPU @ 2.30GHz 2.30GHz với GPU NVIDIA GeForce GTX 1050. Mô hình được huấn luyện trên nền tảng Google Colab với GPU NVIDIA Tesla T4.

7.3 Kết quả và phân tích

7.3.1 Kích thước phân đoạn ω

Kích thước các phân đoạn cũng là một đại lượng có thể ảnh hưởng đến thuật toán, do đó ta sẽ thực hiện khảo sát lần lượt với các kích thước $\omega = (100, 125, 150, 200)$, với ω là chiều dài phân đoạn. Các thông số thiết lập như sau:

- Ngõ vào: Tín hiệu trong miền thời gian & Biến đổi Fourier của tín hiệu trong miền tần số
- Phương pháp trích xuất rPPG: Local variance group (LGI)
- Mô hình: LSTM

Cuối cùng, ta được kết quả như sau:

Bảng 7.1: Khảo sát chiều dài phân đoạn

ω	Training loss	Training acc	Validation loss	Validation acc
100	0.4420	78.56%	0.4767	78.55%
125	0.3752	82.14%	0.4650	81.47%
150	0.2863	87.60%	0.3932	83.73%
200	0.3702	85.03%	0.4420	81.13%

Chiều dài phân đoạn $\omega = 150$ có kết quả huấn luyện tốt nhất với validation loss đạt giá trị nhỏ nhất (0.3932) và đạt validation cao nhất (83.73%). Do đó, đồ án sẽ chọn chiều dài cho các phân đoạn là $\omega = 150$ để thực hiện các khảo sát bên dưới.

7.3.2 Khảo sát phương pháp trích xuất rPPG

Mặc dù phương pháp LGI được chứng minh rằng có hiệu quả tốt hơn so với các phương pháp khác [6]. Đồ án cũng sẽ thực hiện khảo sát so sánh các phương pháp với nhau để kiểm định cho nhận định này. Thực hiện lần lượt so sánh quá trình huấn luyện của 3 phương pháp sau: CHROM, POS và LGI. Với ngõ vào sẽ là tín hiệu trong miền thời gian và biến đổi Fourier trong miền tần số. Ta có kết quả như bảng sau: Như vậy, phương pháp sử dụng LGI đạt độ

Bảng 7.2: So sánh giữa các phương pháp trích xuất rPPG

	CHROM	POS	LGI
Training Acc	73.81%	83.75%	87.60%
Validation Acc	78.23%	81.46%	83.73%

chính xác cao hơn các phương pháp còn lại. Do đó, phương pháp này được chọn để sử dụng cho mô hình. Về mặt lý thuyết, phương pháp LGI đạt độ chính xác cao hơn do các video được sử dụng bị nhiễu do chuyển động gây ra nhiều hơn so với các video được dùng để đo lường nhịp

tim trong y tế. Mà phương pháp LGI tập trung nhiều vào việc giải quyết vấn đề chuyển động hơn hai phương pháp còn lại, do đó, phương pháp này phù hợp hơn trong bài toán nhận dạng video thật/ giả.

7.3.3 Khảo sát đặc trưng đầu vào

Các đặc trưng trong miền không gian và tần số đều mang lại thông tin có ích cho việc phân loại video thật và giả [2]. Ta thực hiện khảo sát độ chính xác Acc cho các đặc trưng này. Thực hiện lần lượt với trường hợp chỉ sử dụng tín hiệu trong miền không gian, biến đổi Fourier trong miền tần số và kết hợp hai loại đặc trưng này.

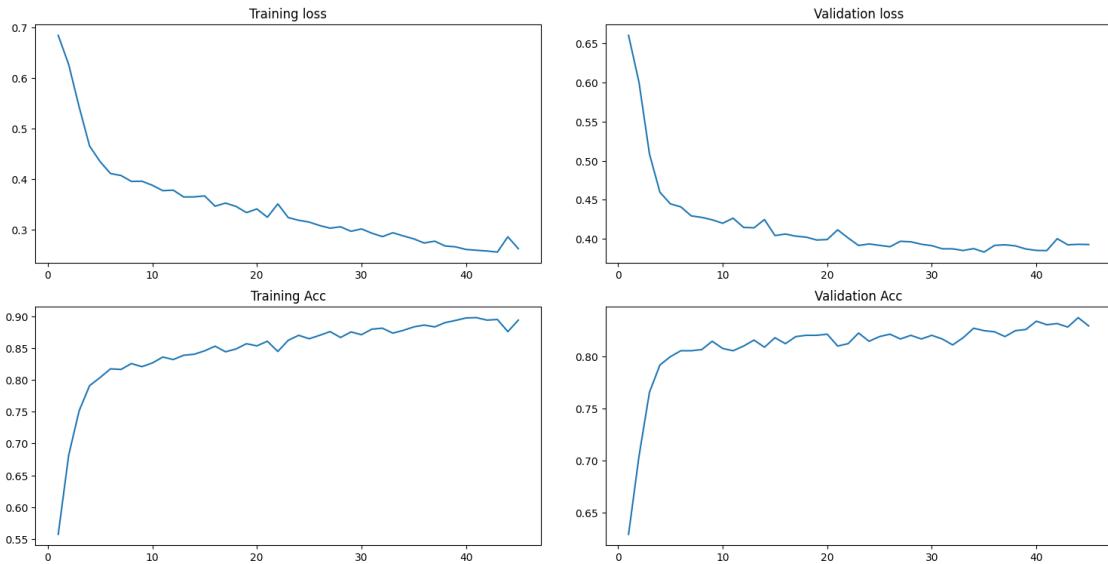
Bảng 7.3: So sánh các loại đặc trưng

Đặc trưng	Training Acc	Validation Acc
Tín hiệu	82.43%	77.82%
Biến đổi Fourier	88.34%	80.59%
Tổng hợp	87.60%	83.73%

Dựa vào Bảng trên, ta nhận thấy chỉ riêng đặc trưng tín hiệu trong miền thời gian hay biến đổi Fourier trong miền tần số đã có khả năng phân lớp dữ liệu. Kết quả cho thấy rằng với các đặc trưng riêng lẻ (tín hiệu, biến đổi Fourier) mô hình đã có khả năng học tập (đạt 77.82% đối với tín hiệu và 80.59% đối với biến đổi Fourier). Khi kết hợp các thông số này lại với nhau ta được kết quả tăng lên đáng kể lần lượt là 87.60% và 83.73% đối với tập huấn luyện và validation. Do đó, ta sử dụng cả hai đặc trưng này làm đặc trưng ngõ vào cho mô hình.

7.3.4 Mô hình LSI+ LSTM

Mô hình được huấn luyện với 46 epochs với kết quả giá trị lỗi (loss) và acc như sau:



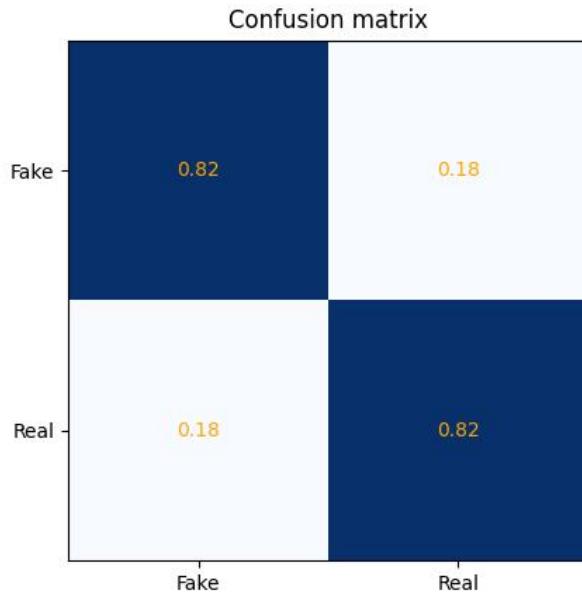
Hình 7.1: Quá trình huấn luyện mô hình

Như Hình 7.1 giá trị lỗi của quá trình huấn luyện và validation đều giảm, tuy nhiên, đến khoảng epoch 80 giá trị lỗi của validation bắt đầu tăng trong khi giá trị lỗi của quá trình huấn luyện vẫn giảm. Khi này, ta dừng việc học của thuật toán bằng kĩ thuật Early Stop để tránh hiện tượng overfitting đối với dữ liệu. Đối với accuracy, cả hai quá trình huấn luyện và validation, giá trị này đều tăng chứng tỏ mô hình đã học được khả năng phân loại đối tượng. Từ quá trình trên, ta thu được kết quả như sau:

Bảng 7.4: Kết quả quá trình huấn luyện

Training loss	Training Acc	Validation loss	Validation Acc
0.2863	87.60%	0.3932	83.73%

Để kiểm tra tính chính xác của thuật toán, đồ án thực hiện việc kiểm thử với một bộ dữ liệu test. Kết quả đạt độ chính xác 82.23%. Cụ thể ta có ma trận nhầm lẫn như sau:



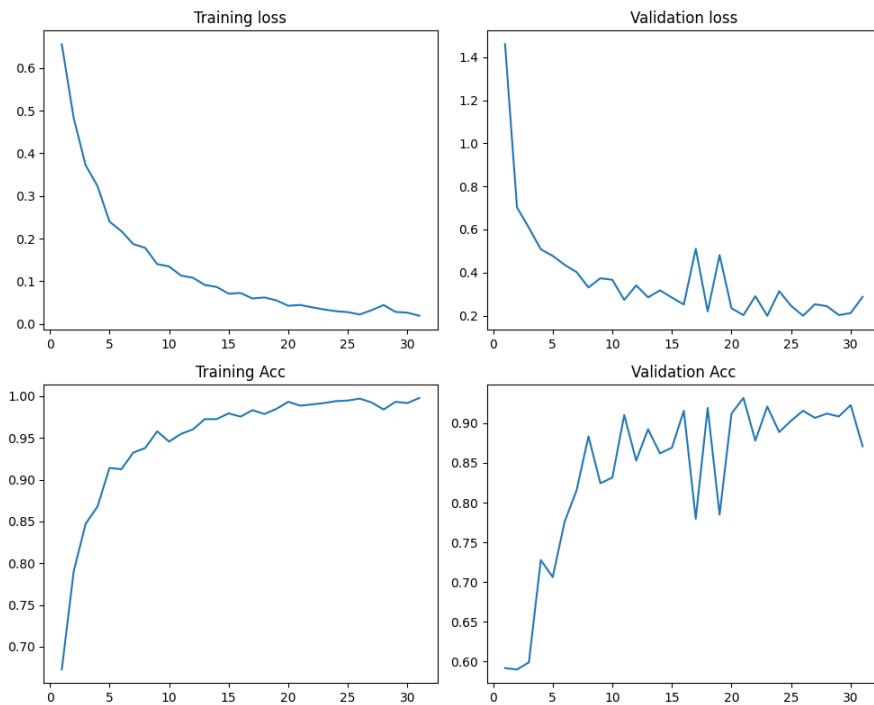
Hình 7.2: Ma trận nhầm lẫn với tập test

Kết quả cho thấy khá tương đồng giữa tỉ lệ True positive và True negative (82% và 82%). Kết quả này ở mức độ tốt tuy nhiên tỉ lệ sai vẫn còn ở mức cao 18% và 18% lần lượt đối với False positive và False negative. Đối với giá trị Precision, Recall, F1-Score, ta lần lượt thu được kết quả như sau:

- Precision = 0.84
- Recall = 0.82
- F1-Score = 0.83

7.3.5 Mô hình LSTM kết hợp với CNN

Đây là mô hình cải tiến của mô hình LGI+LSTM trên với việc kết hợp thêm mạng CNN. Kết quả huấn luyện với đạt như sau:



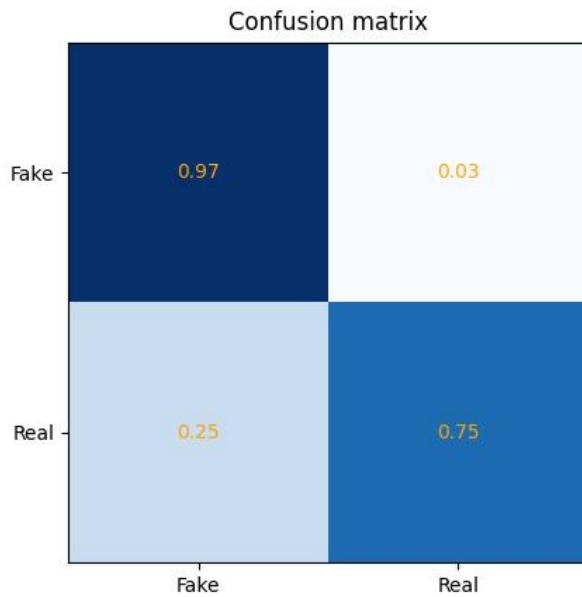
Hình 7.3: Quá trình huấn luyện mô hình LSTM+CNN

Mô hình cho thấy kết quả huấn luyện tốt hơn so với mô hình LSTM+LGI bên trên. Mô hình đạt kết quả độ chính xác trên tập validation đạt 93.19%. Cụ thể như sau:

Bảng 7.5: Kết quả quá trình huấn luyện mô hình LSTM+CNN

Training loss	Training Acc	Validation loss	Validation Acc
0.0445	98.85%	0.2021	93.19%

Tương tự, ta cũng thực hiện khảo sát với tập dữ liệu test và đạt độ chính xác 87.61% . Thực hiện khảo sát ma trận nhầm lẫn ta có kết quả như sau:



Hình 7.4: Ma trận nhầm lẫn với tập test mô hình cải tiến

Ma trận trên cho thấy rằng mô hình có độ chính xác tốt hơn 97% với phân lớp video giả. Tuy nhiên, đối với video thật độ chính xác có giảm đi còn 75%. Đối với Precision, Recall, F1-Score, ta lần lượt thu được kết quả như sau:

- Precision = 0.83
- Recall = 0.97
- F1-Score = 0.89

Như vậy, việc bổ sung mạng CNN vào mô hình đã giúp cải thiện độ chính xác của mô hình. Nhìn chung, cả hai mô hình có thể sử dụng được trong đa số các trường hợp, tuy nhiên, cần cải thiện thêm về mặt độ chính xác.

7.3.6 So sánh mô hình

Cuối cùng để đánh giá một cách khách quan, đồ án cũng thực hiện lại mô hình [2] và [1]. Sau đó, thực hiện việc so sánh kết quả với hai mô hình này. Trước tiên, ta sẽ tìm hiểu một cách tổng quát về các mô trình trên:

Như ở bảng trên, cột đầu tiên là tên của bài báo/ phương pháp, cột thứ hai (Trích xuất rPPG) chỉ phương pháp được dùng để trích xuất rPPG, cột thứ ba (Mô hình) dùng để chỉ phương pháp được huấn luyện để phân lớp dữ liệu, cột cuối cùng (Các đặc trưng) chỉ các thông số đầu vào của mô hình. Từ đó, ta thực hiện và thu được kết quả như sau:

Như vậy, so với mô hình [2] và [1], mô hình sử dụng LSTM+CNN trong đồ án đạt kết quả huấn luyện cao nhất là 93.19%. Mô hình LGI+LSTM (83.73%) cho kết quả cao hơn mô hình

Bảng 7.6: So sánh mô hình với các phương pháp khác

	Trích xuất rPPG	Mô hình	Các đặc trưng
Fakecatcher [2]	CHROM	CNN	- Tín hiệu - Mật độ phổ công suất (PSD)
SVM [1]	POS	SVM	- Thông số đánh giá độ phức tạp của tín hiệu - Thông số đo lường sự nhất quán
LGI+LSTM	LGI	LSTM	- Tín hiệu - Biến đổi Fourier
CNN+LSTM	LGI	LSTM CNN	- Tín hiệu - Biến đổi Fourier - Khuôn mặt

Bảng 7.7: So sánh hiệu quả với các phương pháp khác

Mô hình	Training loss	Training Acc	Validation loss	Validation Acc
Fakecatcher [2]	0.5211	84.15%	0.5362	84.22%
SVM [1]	—	—	—	75.68%
LGI+LSTM	0.2863	87.60%	0.3932	83.73%
LSTM+CNN	0.0445	98.85%	0.2021	93.19%

SVM (75.68%) nhưng lại thấp hơn mô hình Fakecatcher (84.22%). Như vậy, từ kết quả trên ta có thể kết luận như sau:

- Mô hình CNN đạt hiệu quả cao hơn so với mô hình sử dụng LSTM hay SVM.
- Việc kết hợp giữa tín hiệu rPPG và bìa ngoài (khuôn mặt) đem lại kết quả độ chính xác tốt hơn.

7.3.7 So sánh với các dạng video Deepfake

Để đảm bảo tính khách quan, đồ án tốt nghiệp này thực hiện so sánh kết quả với các mô hình tạo video Deepfake khác nhau từ 10 công cụ video khác nhau. Các công cụ này có điểm khác nhau giữa yêu cầu đầu vào. Cụ thể, có thể chia thành 2 nhóm như sau:

- Ngõ vào gồm 2 video: 1 video mẫu và 1 video của đối tượng được làm giả.
- Ngõ vào gồm 1 video mẫu và 1 ảnh của đối tượng được làm giả.

Hình bên dưới sẽ minh họa ngõ ra của các ứng dụng/ mô hình được sử dụng cho mục đích so sánh:

Như Hình 7.5, các mô hình MyHeritage, Revive, TalkingPhoto, ROOP, Swapface, Thin-splat motion là các mô hình yêu cầu ngõ vào bao gồm 1 video và 1 ảnh. Các mô hình còn lại Deepfake, Face2face, Faceswap, NeuralTexture yêu cầu 2 video làm dữ liệu đầu vào. Ngõ ra, giữa các video cũng có sự khác biệt: các mô hình như MyHeritage, TalkingPhoto, Thin-splat motion có ngõ ra bị giảm độ phân giải và ít chân thực hơn các video tạo từ các công cụ khác.



Hình 7.5: So sánh giữa các mô hình tạo video Deepfake

Kết quả thử nghiệm, mô hình không hiệu quả đối với các video được tạo bởi 2 video do độ phức tạp của video cao hơn, khác biệt nhiều so với tập dữ liệu huấn luyện. Ngoài ra, cũng có thể lý giải rằng do khi này độ chân thực của các video giả đã tăng lên dẫn đến cấu trúc mạng CNN không còn hiệu quả nữa.

Đối với các mô hình/ ứng dụng sử dụng 1 video và 1 ảnh kết quả cho ra tương đối tốt. Cụ thể đối với video được tạo bởi Revive đạt độ chính xác 76.67% (50 video test), MyHeritage đạt độ chính xác 93.33% (30 video test), TalkingPhoto đạt độ chính xác 90.0% (30 video test),

7.4 Kết luận chương

Kết quả cho thấy phương pháp sử dụng của đồ án có hiệu quả tốt với độ chính xác 93.19% trên tập validation và 87.61% với tập dữ liệu test. Ngoài ra, mô hình cũng cho thấy hiệu quả hơn so với các mô hình đã được nghiên cứu trước đó. Đối với các dạng video giả thực tế, mô hình đạt hiệu suất tốt với các mô hình được tạo bởi 1 video và 1 ảnh, trong khi đó, các mô hình tạo bằng 2 video mô hình còn nhiều hạn chế và cần được nghiên cứu thêm

Chương 8

KẾT LUẬN

Chương này sẽ tổng kết những kết quả đã đạt được từ Chương 7. Từ đó, đồ án sẽ tổng hợp những điều đạt được và những hạn chế còn thiếu sót ở đồ án. Cuối cùng, đồ án cũng đề xuất các hướng phát triển cho đề tài.

8.1 Tóm tắt và kết luận chung

8.1.1 Những đóng góp của đề tài

Trong đồ án này, nhận dạng video giả được thực hiện dựa vào tín hiệu sinh học rPPG (tín hiệu này phát sinh bởi sự thay đổi lưu lượng máu của do sự co bóp của nhịp tim) kết hợp với các mô hình máy học để thực hiện việc phân loại 2 lớp (thật/ giả). Đồ án đã thực hiện các phần sau:

- Một là phần khảo sát về cơ sở lý thuyết về các phương pháp xử lý tín hiệu, các phương pháp trích xuất remote photoplethysmography và các mô hình máy học. Ngoài ra, đồ án cũng thực hiện tổng hợp các nghiên cứu, công trình trước đó. Trong phần cơ sở lý thuyết này, đặc biệt nhấn mạnh đến phương pháp trích xuất LGI, mô hình recurrent neural network, LSTM và kĩ thuật Deep supervision.
- Hai là phần ứng dụng của các cơ sở lý thuyết trên. Trong phần này, đồ án đã xây dựng mô hình LSTM sử dụng Deep supervision. Tín hiệu rPPG được trích xuất từ phương pháp LGI được dùng làm ngõ vào của mô hình. Ngoài ra, đồ án cũng kết hợp với mô hình CNN để cải thiện hiệu suất của mô hình. Trong phần này, đồ án cũng khảo sát các thông số, độ chính xác và hiệu quả của mô hình. Cuối cùng, đồ án thực hiện việc khảo sát với các mô hình khác sử dụng Support vector machine (SVM) và Convolutional neural network (CNN).

Kết quả thuật toán đã sử dụng LSTM để khảo sát tính hiệu quả của tín hiệu trong nhận dạng video giả. Kết quả cho thấy LSTM có hiệu quả cao hơn với độ chính xác đạt 82.23% và mô hình cải tiến đạt 87.61%. Nhìn chung, đồ án đã đạt được những giả thuyết đã đặt ra.

8.1.2 Những hạn chế

- Dữ liệu được huấn luyện cho mô hình chưa đủ lớn do đó, độ chính xác chưa thể đạt được mức cao hơn. Cụ thể, thuật toán hoạt động tốt với các trường hợp khuôn mặt ở chính diện không bị che khuất bởi bất kì vật gì. Còn đối với các trường hợp như bên dưới, mô hình thường không phân biệt đúng giữa video thật và giả.

- Mô hình còn bị hạn chế ở các video được huấn luyện tạo bởi 2 video khác nhau. Do tập huấn luyện giữa đồ án và dữ liệu do 2 video tạo ra có sự khác biệt lớn.

- Thuật toán gặp hạn chế ở các video mặt người mà khu vực ROI (vùng má, mũi) bị che khuất bởi các đồ vật như mắt kính, râu,... Các trường hợp bị che khuất một phần khuôn mặt cũng gây ra sự nhầm lẫn của thuật toán. Nguyên nhân chủ yếu của các lỗi này bắt nguồn từ việc nhận dạng vùng ROI bị lỗi, do đó, các khắc phục lỗi này được đề xuất bằng việc sử dụng các công cụ nhận dạng khuôn mặt mạnh mẽ hơn như: MTCNN, Mediapipe, Retinaface,... Việc bổ sung thêm các tập dữ liệu về mặt người bị che khuất cũng có thể có ích trong trường hợp này.

- Ngoài ra, đồ án chỉ hướng đến dữ liệu video chỉ có 1 người mà chưa xem xét đến video có nhiều người.

- Việc tìm kiếm nguồn dữ liệu về rPPG để phát triển thuật toán để trích xuất tín hiệu rPPG cũng là một thách thức lớn. Đây là yếu tố mấu chốt ảnh hưởng đến độ chính xác của mô hình. Ngoài ra, đồ án đặt giả thuyết rằng mô hình cải tiến nếu được sử dụng với mạng Convolutional-LSTM sẽ đạt kết quả cao hơn nhưng do hạn chế về dung lượng RAM trên Google Colab nên phương pháp này chưa được thử nghiệm.

8.2 Hướng phát triển

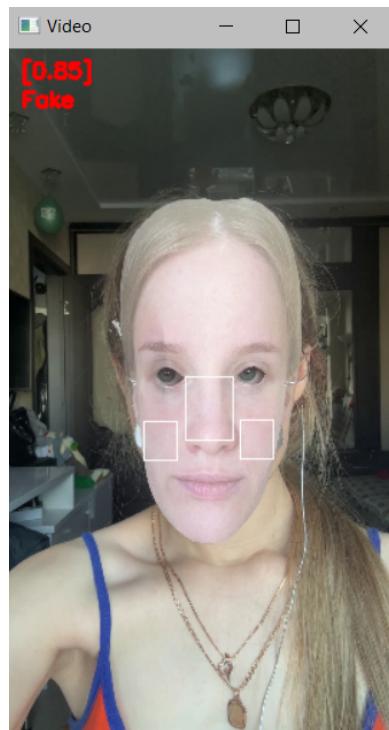
Để cải thiện độ chính xác của mô hình, một số đề xuất để phát triển thuật toán như sau:

- Các thuật toán trích xuất rPPG bằng các phương pháp xử lý ảnh mặc dù có tốc độ xử lý nhanh. Tuy nhiên, các phương pháp này rất dễ bị nhiễu. Do đó, có thể xem xét việc cải thiện thuật toán trích xuất tín hiệu rPPG, các thuật toán mới như [18] có thể đạt hiệu quả cao hơn.
- Kết hợp với các đặc trưng được thể hiện: background, mắt, miệng,... để đạt được hiệu quả cao hơn.

Trên thực tế, mô hình này không chỉ hiệu quả trong việc nhận dạng video thật/ giả giúp ích cho một số ứng dụng như sau:

- Ứng dụng trong định danh khuôn mặt (Liveness detection) trong các giao dịch, định danh điện tử,...
- Lọc tin giả, video giả trên các nền tảng video call, mạng xã hội

- Phát hiện người đeo mặt nạ, người có ý định tấn công (Presentation Attack Detection)



Hình 8.1: Video Presentation Attack Detection

Đồ án tốt nghiệp cũng thử nghiệm trên 10 video PAD. Kết quả cũng đạt được 7/10 video nhận dạng đúng. Do với video đeo mặt nạ không có sự thay đổi mức xám, điều này khá giống với dạng video giả mà mô hình đang xét tới. Như vậy, nếu mô hình được huấn luyện thì việc phát hiện Presentation Attack Detection là hoàn toàn có thể thực hiện được.

Tài liệu tham khảo

- [1] Giuseppe Boccignone, Sathya Bursic, Vittorio Cuculo, Alessandro D'Amelio, Giuliano Grossi, Raffaella Lanzarotti, and Sabrina Patania. *DeepFakes Have No Heart: A Simple rPPG-Based Method to Reveal Fake Videos*, pages 186–195. 05 2022.
- [2] Umur Aybars Ciftci, Ilke Demir, and Lijun Yin. Fakematcher: Detection of synthetic portrait videos using biological signals. *IEEE transactions on pattern analysis and machine intelligence*, 2020.
- [3] Umur Aybars Ciftci, Ilke Demir, and Lijun Yin. How do the hearts of deep fakes beat? deep fake source detection via interpreting residuals with biological signals. In *2020 IEEE international joint conference on biometrics (IJCB)*, pages 1–10. IEEE, 2020.
- [4] Gerard De Haan and Vincent Jeanne. Robust pulse rate from chrominance-based rppg. *IEEE Transactions on Biomedical Engineering*, 60(10):2878–2886, 2013.
- [5] Steven Fernandes, Sunny Raj, Eddy Ortiz, Iustina Vintila, Margaret Salter, Gordana Urosevic, and Sumit Jha. Predicting heart rate variations of deepfake videos using neural ode. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 1721–1729, 2019.
- [6] Fridolin Haugg, Mohamed Elgendi, and Carlo Menon. Effectiveness of remote ppg construction methods: A preliminary analysis. *Bioengineering*, 9(10), 2022.
- [7] Tackhyun Jung, Sangwon Kim, and Keecheon Kim. Deepvision: Deepfakes detection using human eye blinking pattern. *IEEE Access*, 8:83144–83154, 2020.
- [8] Dae-Yeol Kim, Kwangkee Lee, and Chae-Bong Sohn. Assessment of roi selection for facial video-based rppg. *Sensors*, 21(23):7923, 2021.
- [9] Seung-Hyun Kim, Su-Min Jeon, and Eui Chul Lee. Face biometric spoof detection method using a remote photoplethysmography signal. *Sensors*, 22(8):3070, 2022.
- [10] Sungjun Kwon, Jeehoon Kim, Dongseok Lee, and Kwangsuk Park. Roi analysis for remote photoplethysmography on facial video. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 4938–4941. IEEE, 2015.

- [11] Renjie Li, Xinyi Wang, Guan Huang, Wenli Yang, Kaining Zhang, Xiaotong Gu, Son N Tran, Saurabh Garg, Jane Alty, and Quan Bai. A comprehensive review on deep supervision: Theories and applications. *arXiv preprint arXiv:2207.02376*, 2022.
- [12] Xin Liu, Josh Fromm, Shwetak Patel, and Daniel McDuff. Multi-task temporal shift attention networks for on-device contactless vitals measurement. *Advances in Neural Information Processing Systems*, 33:19400–19411, 2020.
- [13] Momina Masood, Marriam Nawaz, Khalid Malik, Ali Javed, Aun Irtaza, and Hafiz Malik. Deepfakes generation and detection: state-of-the-art, open challenges, countermeasures, and way forward. *Applied Intelligence*, 53:1–53, 06 2022.
- [14] Falko Matern, Christian Riess, and Marc Stamminger. Exploiting visual artifacts to expose deepfakes and face manipulations. *2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, pages 83–92, 2019.
- [15] Christian S Pilz, Sebastian Zaunseder, Jarek Krajewski, and Vladimir Blazek. Local group invariance for heart rate estimation from face videos in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 1254–1262, 2018.
- [16] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics++: Learning to detect manipulated facial images, 2019.
- [17] Aliaksandr Siarohin, Oliver J. Woodford, Jian Ren, Menglei Chai, and Sergey Tulyakov. Motion representations for articulated animation. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13648–13657, 2021.
- [18] Jeremy Speth, Nathan Vance, Benjamin Sporrer, Lu Niu, Patrick Flynn, and Adam Czajka. Hallucinated heartbeats: Anomaly-aware remote pulse estimation. *arXiv preprint arXiv:2303.06452*, 2023.
- [19] Wim Verkruyse, Lars Svaasand, and J Nelson. Remote plethysmographic imaging using ambient light. *Optics express*, 16:21434–45, 12 2008.
- [20] Wenjin Wang, Albertus C Den Brinker, Sander Stuijk, and Gerard De Haan. Algorithmic principles of remote ppg. *IEEE Transactions on Biomedical Engineering*, 64(7):1479–1491, 2016.
- [21] Wenjin Wang, Albertus C. den Brinker, Sander Stuijk, and Gerard de Haan. Algorithmic principles of remote ppg. *IEEE Transactions on Biomedical Engineering*, 64(7):1479–1491, 2017.
- [22] Pu Sun Honggang Qi Yuezun Li, Xin Yang and Siwei Lyu. Celeb-df: A large-scale challenging dataset for deepfake forensics. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

- [23] Jian Zhao and Hui Zhang. Thin-plate spline motion model for image animation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3647–3656, 2022.

Phụ lục A

CODE THỰC HIỆN

A.1 Code trích xuất rPPG:

Biến đổi Fourier:

```
FFT = [[ ], [ ], [ ]]
for i in range(signal.shape[0]):
    FFT[i].append(abs(fft.fft(signal[i])))
FFT = np.array(FFT).reshape(3,150)
return FFT
```

Tiền xử lý frame:

```
RGB = []
for frame in frames:
    summation = np.sum(np.sum(frame, axis=0), axis=0)
    RGB.append(summation / (frame.shape[0] * frame.shape[1]))
RGB = np.asarray(RGB)
RGB = RGB.transpose(1, 0).reshape(1, 3, -1)
```

Trích xuất rPPG:

```
precessed_data = process_video(frames)
U, _, _ = np.linalg.svd(precessed_data)
S = U[:, :, 0]
S = np.expand_dims(S, 2)
SST = np.matmul(S, np.swapaxes(S, 1, 2))
p = np.tile(np.identity(3), (S.shape[0], 1, 1))
P = p - SST
Y = np.matmul(P, precessed_data)
bvp = Y[:, 1, :]
```

```
bvp = bvp.reshape(-1)
```

A.2 Mô hình huấn luyện:

Code mô hình gồm 2 mô hình con:

- rPPG_model: mạng LSTM sử dụng rPPG làm đặc trưng
- Appear_model: mạng CNN sử dụng khuôn mặt làm đặc trưng

```
def rPPG_model():  
    weight_decay = 0.0001  
    inputs0 = Input(shape=(150, 3))  
    inputs1 = Input(shape=(150, 3))  
    in0 = Bidirectional(LSTM(100))(inputs0)  
    in0 = Dense(50,kernel_regularizer=regularizers.l2(weight_decay))(in0)  
    in0 = Activation('leaky_relu')(in0)  
    in1 = Bidirectional(LSTM(100))(inputs1)  
    in1 = Dense(50,kernel_regularizer=regularizers.l2(weight_decay))(in1)  
    in1 = Activation('leaky_relu')(in1)  
    merged = Concatenate()([in0, in1])  
    merged = Dense(128,kernel_regularizer=regularizers.l2(weight_decay))(merged)  
    merged = Activation('leaky_relu')(merged)  
    merged = Dropout(0.5)(merged)  
    output = Dense(1)(merged)  
    output = Activation('sigmoid',name='Sout1')(output)  
    inputs = [inputs0,inputs1]  
    outputs = [output]  
    model = Model(inputs, outputs)  
    return model  
  
def inception_module(x, filters):  
    weight_decay = 0.0005  
    branch1x1 = Conv2D(filters[0], (1, 1), padding='same',  
                      activation='relu',kernel_regularizer=regularizers.l2(weight_decay))(x)  
    branch1x1 = Dropout(0.3)(branch1x1)  
    branch3x3 = Conv2D(filters[1], (1, 1), padding='same',  
                      activation='relu',kernel_regularizer=regularizers.l2(weight_decay))(x)  
    branch3x3 = Conv2D(filters[2], (3, 3), padding='same',  
                      activation='relu',kernel_regularizer=regularizers.l2(weight_decay))(branch3x3)  
    branch3x3 = Dropout(0.3)(branch3x3)
```

```

branch5x5 = Conv2D(filters[3], (1, 1), padding='same',
activation='relu',kernel_regularizer=regularizers.l2(weight_decay))(x)
branch5x5 = Conv2D(filters[4], (5, 5), padding='same',
activation='relu',kernel_regularizer=regularizers.l2(weight_decay))(branch5x5)
branch5x5 = Dropout(0.3)(branch5x5)
branch_pool = MaxPooling2D((3, 3), strides=(1, 1), padding='same')(x)
branch_pool = Conv2D(filters[5], (1, 1), padding='same',
activation='relu',kernel_regularizer=regularizers.l2(weight_decay))(branch_pool)
output = Concatenate(axis=-1)([branch1x1, branch3x3, branch5x5, branch_pool])
return output

def Appear_model():
    weight_decay = 0.0005
    inputs2 = Input(shape=(100,150,100))
    x = Conv2D(64, (7, 7), strides=(2, 2),
padding='same',kernel_regularizer=regularizers.l2(weight_decay))(inputs2)
    x = BatchNormalization()(x)
    x = Activation('relu')(x)
    x = Dropout(0.3)(x)
    x = MaxPooling2D((3, 3), strides=(2, 2), padding='same')(x)
    x = Conv2D(64, (3, 3), padding='same',kernel_regularizer=regularizers.l2(weight_decay))(x)
    x = BatchNormalization()(x)
    x = Activation('relu')(x)
    x = Dropout(0.3)(x)
    x = MaxPooling2D((3, 3), strides=(2, 2), padding='same')(x)
    x = inception_module(x, [16, 24, 32, 8, 8, 8])
    x = AveragePooling2D((7, 7), strides=(1, 1))(x)
    x = Flatten()(x)
    x = Dense(64,kernel_regularizer=regularizers.l2(weight_decay))(x)
    x = Activation('leaky_relu')(x)
    x = Dropout(0.3)(x)
    sub_out2 = Dense(1)(x)
    sub_out2 = Activation('sigmoid', name="Sout2")(sub_out2)
    inputs = [inputs2]
    outputs = [sub_out2]
    model = Model(inputs, outputs)
    return model

loss = tf.keras.losses.BinaryCrossentropy()
opt = optimizers.Adam(1e-4, decay=1e-5)
metrics = ['accuracy',tf.keras.metrics.AUC()]

```

```
early_stop = EarlyStopping(monitor='val_loss',patience=5, verbose=1)
modelA = rPPG_model()
modelA.compile(loss=loss,optimizer=opt, metrics=metrics)
modelB = Appear_model()
modelA.compile(loss=loss,optimizer=opt, metrics=metrics)
weight_decay = 0.0005 output = Concatenate()([modelA.layers[-4].output,modelB.layers[-4].output])
output = Dense(128,kernel_regularizer=regularizers.l2(weight_decay))(output)
output = BatchNormalization()(output)
output = Activation('leaky_relu')(output)
output = Dropout(0.5)(output)
output = Dense(1)(output)
output = Activation('sigmoid',name='f_out')(output)
model = Model(inputs = [modelA.input, modelB.input], outputs = [modelA.output,
modelB.output,output])
```