

TRƯỜNG ĐẠI HỌC THỦY LỢI
PHÂN HIỆU



ĐỀ TÀI:

**Ứng dụng xử lý tín hiệu & Machine Learning/Deep Learning để phân tích ECG
12-lead, hỗ trợ phát hiện các bệnh tim mạch**

Sinh viên thực hiện: Lê Thị Yến Nhi

Đào Hoàng Mỹ Tiên

Lớp: S25-64CNTT

Bộ môn: Công Nghệ Thông Tin

Giáo viên hướng dẫn: ThS. Vũ Thị Hạnh

Hồ Chí Minh, 2025

MỤC LỤC

MỤC LỤC	2
DANH MỤC HÌNH ẢNH	4
DANH MỤC CÁC BẢNG	4
I. MỞ ĐẦU	7
1. Tổng quan tình hình nghiên cứu	7
2. Mục tiêu đề tài	7
3. Phương pháp, đối tượng và phạm vi nghiên cứu	8
3.1. Phương pháp nghiên cứu	8
3.2. Đối tượng nghiên cứu	8
3.3. Phạm vi nghiên cứu	8
II. NỘI DUNG VÀ KẾT QUẢ NGHIÊN CỨU	10
Chương 1: Giới thiệu tổng quan về điện tâm đồ 12 chuyển đạo (ECG 12 Leads)10	
1.1. Khái niệm về điện tâm đồ (ECG)	10
1.2. Cấu trúc và nguyên lý hoạt động	10
1.2.1. Sơ đồ điện cực và các chuyển đạo	10
1.2.2. Cơ chế hoạt động	12
1.3. Cấu trúc sóng điện tâm đồ và ý nghĩa lâm sàng	12
1.4. Ứng dụng của ECG 12 chuyển đạo trong chẩn đoán bệnh lý tim mạch13	
1.5. Đặc điểm ECG theo chuyển đạo của 8 nhãn bệnh	14
1.6. Cơ sở cho việc áp dụng học sâu và giải thích mô hình trên ECG	20
Chương 2: Mô hình ResNet và Inception Time	21
2.1. Tổng quan về học sâu trong phân tích tín hiệu ECG	21
2.2. Kiến trúc ResNet và phiên bản ResNet1D-CNN	21
2.2.1. Mạng ResNet là gì?	21
2.2.2. ResNet1D-CNN cho tín hiệu ECG	23
2.3. InceptionTime cho tín hiệu ECG	24
2.4. Giải thích mô hình bằng Attention	25
Chương 3: Thu thập và tiền xử lý dữ liệu	27
3.1. Thu thập dữ liệu	27

3.2. Tiền xử lí dữ liệu	27
Chương 4: Xây dựng và huấn luyện mô hình	34
4.1. Xây dựng mô hình ResNet 1D-CNN	34
4.2. Xây dựng mô hình InceptionTime	40
4.3. Huấn luyện và đánh giá mô hình	41
4.4. Đánh giá mô hình với Attention	44
4.5. Kết quả mô hình	45
Chương 5: Xây dựng Web App nhận diện và dự đoán bệnh từ tín hiệu điện tâm đồ	46
5.1. Xây dựng FrontEnd và BackEnd	46
5.2. Cấu hình để load mô hình nhận diện sóng điện tâm đồ	47
III. TỔNG KẾT	49
1. Kết luận chung về đề tài	49
2. Hướng phát triển của đề tài	49
TÀI LIỆU THAM KHẢO	51

DANH MỤC HÌNH ẢNH

Hình 1 : Minh họa quá trình đo điện tâm đồ	10
Hình 2 : Vị trí đo của các chuyển đạo	11
Hình 3 : Minh họa đo chuyển đạo I, II, III	11
Hình 4 : Nhận biết các sóng P, PR, ST, QRS, QT, T	13
Hình 5 : Công thức vẽ lưới ECG	14
Hình 6 : Minh họa lưới ECG của Nhịp chậm xoang	15
Hình 7 : Minh họa sóng bệnh Cường nhĩ	15
Hình 8 : Minh họa sóng ST chênh lên	16
Hình 9 : So sánh giữa nhịp xoang bình thường và nhịp chậm xoang	17
Hình 10 : So sánh sóng của người bình thường, cường nhĩ và rung nhĩ ...	18
Hình 11 : Minh họa sóng ST chênh xuống	18
Hình 12 : Minh họa sóng trục điện tim lệch trái	19
Hình 13 : Tỷ lệ training error	22
Hình 14 : Kỹ thuật Skip Connection	23
Hình 15 : Kiến trúc mô hình InceptionTime	24
Hình 16 : Cấu trúc thư mục dữ liệu huấn luyện	27
Hình 17 : Nội dung file dữ liệu bệnh nhân	28
Hình 18 : Nội dung file chứa dữ liệu mã bệnh đã được mapping	29
Hình 19 : Top 10 nhãn bệnh phổ biến	29
Hình 20 : Hình ảnh sóng thu được sau chuẩn hóa	31
Hình 21 : Xây dựng pipeline huấn luyện dữ liệu	37
Hình 22 : Bảng kết quả classification report ResNet50	41
Hình 23 : Bảng kết quả classification report ResNet50 sau khi chỉnh threshold	42
Hình 24 : Bảng kết quả classification report InceptionTime	43
Hình 25 : Bảng kết quả classification report Inception Time sau khi chỉnh threshold	44
Hình 26 : Minh họa trực quan lại đặc trưng với Attention	45
Hình 27 : Giao diện Web App	46
Hình 28 : Giao diện dự đoán kết quả đa nhãn	47

Hình 29 : Giao diện hiển thị kết quả ảnh 12 chuyển đạo	47
--	----

DANH MỤC CÁC BẢNG

Bảng 1 : Bảng mã dùng để nhận diện	14
Bảng 2 : Phân biệt mức chênh lệch độ đo theo giới tính cho các lead	16
Bảng 3 : Chia lead theo vùng	19
Bảng 4 : Dùng lead II để xác định độ lệch trái	20

I. MỞ ĐẦU

1. Tổng quan tình hình nghiên cứu

Bệnh lý tim mạch là một trong những nguyên nhân hàng đầu gây tử vong và tàn tật trên toàn cầu. Việc phát hiện sớm và chính xác các bất thường về hoạt động điện của tim có ý nghĩa quan trọng trong công tác phòng ngừa và điều trị bệnh. Trong đó, điện tâm đồ 12 chuyển đạo (ECG 12-lead) là phương pháp cận lâm sàng phổ biến, đơn giản và không xâm lấn, thường được sử dụng để ghi lại tín hiệu điện học của tim từ nhiều góc nhìn khác nhau, cung cấp thông tin toàn diện về cấu trúc và chức năng tim.

Tuy nhiên, việc phân tích tín hiệu ECG vẫn chủ yếu dựa vào chuyên môn và kinh nghiệm của bác sĩ, dẫn đến những hạn chế về mặt thời gian, chi phí và khả năng bao quát khi số lượng bệnh nhân lớn. Trong những năm gần đây, các phương pháp học sâu đã thể hiện tiềm năng vượt trội trong việc tự động phân tích tín hiệu sinh lý, đặc biệt là với các bài toán phân loại rối loạn nhịp tim hoặc phát hiện bất thường trên ECG.

Song song với hiệu quả của mô hình, khả năng giải thích (explainability) là một yếu tố quan trọng để các hệ thống trí tuệ nhân tạo có thể được chấp nhận và tin tưởng trong môi trường lâm sàng. Do đó, các phương pháp trực quan hóa như Grad-CAM ngày càng được áp dụng để làm nổi bật các vùng tín hiệu quan trọng mà mô hình đã sử dụng để đưa ra dự đoán. Đây là một bước tiến quan trọng nhằm tăng độ minh bạch và khả năng kiểm chứng của mô hình, tạo điều kiện cho bác sĩ dễ dàng đánh giá, đối chiếu với chuyên môn.

2. Mục tiêu đề tài

Đề tài hướng tới các mục tiêu chính sau:

- Xây dựng mô hình học sâu ResNet1D-CNN có khả năng phân loại tín hiệu ECG 12-lead thành 8 loại bệnh tim phổ biến, được chọn lọc từ tập dữ liệu lớn trên nền tảng PhysioNet.
- Tích hợp cơ chế giải thích Grad-CAM vào mô hình nhằm làm rõ những vùng tín hiệu có ảnh hưởng lớn đến quyết định dự đoán, từ đó tạo điều kiện cho các bác sĩ hiểu rõ hơn về cách mô hình hoạt động.

- Triển khai một giao diện web thân thiện, cho phép người dùng tải lên tín hiệu ECG, quan sát kết quả phân loại, cũng như hình ảnh trực quan hóa dựa trên Grad-CAM.
- Đánh giá hiệu suất và độ tin cậy của mô hình thông qua các chỉ số đánh giá (Precision, Recall, F1-score, AUC) và khảo sát phản hồi từ chuyên gia y tế.

3. Phương pháp, đối tượng và phạm vi nghiên cứu

3.1. Phương pháp nghiên cứu

- Tiền xử lý tín hiệu ECG 12-lead, bao gồm lọc nhiễu, chuẩn hóa biên độ, và xử lý mất cân bằng lớp.
- Xây dựng kiến trúc mô hình ResNet1D-CNN, huấn luyện và tối ưu trên tập huấn luyện, đánh giá trên tập kiểm thử.
- Trích xuất đặc trưng và trực quan hóa bằng phương pháp Grad-CAM, xác định các vùng đóng góp mạnh cho từng nhãn bệnh.
- Xây dựng web-app cho phép người dùng sử dụng mô hình một cách trực quan.
- Khảo sát ý kiến chuyên gia y tế để đánh giá hiệu quả sử dụng thực tế của mô hình và giao diện.

3.2. Đối tượng nghiên cứu

Đối tượng nghiên cứu chính là:

- Tín hiệu điện tâm đồ 12 chuyển đạo được trích xuất từ tập dữ liệu công khai “ECG Arrhythmia Dataset” trên nền tảng PhysioNet, bao gồm hàng chục nghìn bản ghi, mỗi bản kéo dài 10 giây, được gán nhãn theo chuẩn lâm sàng.
- Trong số 64 nhãn bệnh có trong tập dữ liệu, đề tài lựa chọn ra 8 nhãn phổ biến và có giá trị chẩn đoán cao, tiêu biểu như: cuồng nhĩ (AF), nhịp chậm xoang (SB), ST chênh (ST), v.v.

3.3. Phạm vi nghiên cứu

- Nghiên cứu giới hạn trong phạm vi nhận diện 8 nhãn bệnh phổ biến từ tín hiệu ECG 12-lead, không mở rộng ra các dạng tín hiệu khác như ECG Holter, hoặc các dữ liệu hình ảnh chẩn đoán khác.

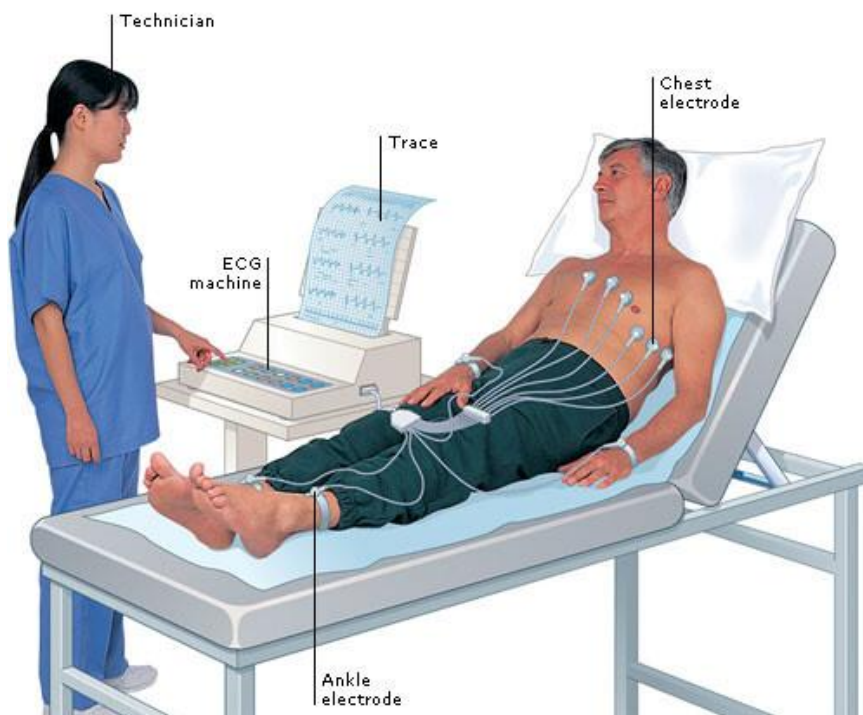
- Ứng dụng web được triển khai ở mức thử nghiệm nguyên mẫu (prototype), chưa tích hợp với hệ thống thông tin bệnh viện hoặc thiết bị phần cứng thực tế.
- Các mô hình được đánh giá trên tập kiểm thử độc lập, chưa áp dụng trên bệnh nhân thật hoặc thử nghiệm lâm sàng.

II. NỘI DUNG VÀ KẾT QUẢ NGHIÊN CỨU

Chương 1: Giới thiệu tổng quan về điện tâm đồ 12 chuyển đạo (ECG 12 Leads)

1.1. Khái niệm về điện tâm đồ (ECG)

Điện tâm đồ (Electrocardiogram – ECG) là phương pháp ghi lại hoạt động điện học của tim thông qua các điện cực đặt trên bề mặt cơ thể. Tín hiệu ECG phản ánh các quá trình khử cực và tái cực của cơ tim trong chu kỳ tim, từ đó giúp bác sĩ đánh giá tình trạng dẫn truyền điện tim, rối loạn nhịp, và các tổn thương cơ tim.



Hình 1: Minh họa quá trình đo điện tâm đồ

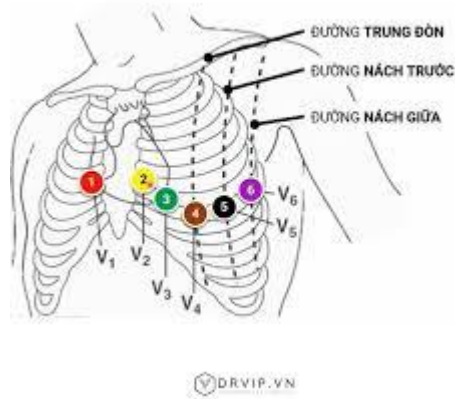
ECG là một xét nghiệm phổ biến, đơn giản, không xâm lấn, có vai trò thiết yếu trong thực hành lâm sàng tim mạch. Nhờ khả năng phản ánh trực tiếp hoạt động điện sinh lý của tim, ECG thường được sử dụng để chẩn đoán nhanh chóng và chính xác các bất thường như: rung nhĩ, ngoại tâm thu, block nhánh, nhồi máu cơ tim cấp, hội chứng QT dài, v.v.

1.2. Cấu trúc và nguyên lý hoạt động

1.2.1. Sơ đồ điện cực và các chuyển đạo

ECG 12-lead bao gồm tổng cộng 12 chuyển đạo (leads), được hình thành từ 10 điện cực đặt tại các vị trí cố định trên cơ thể:

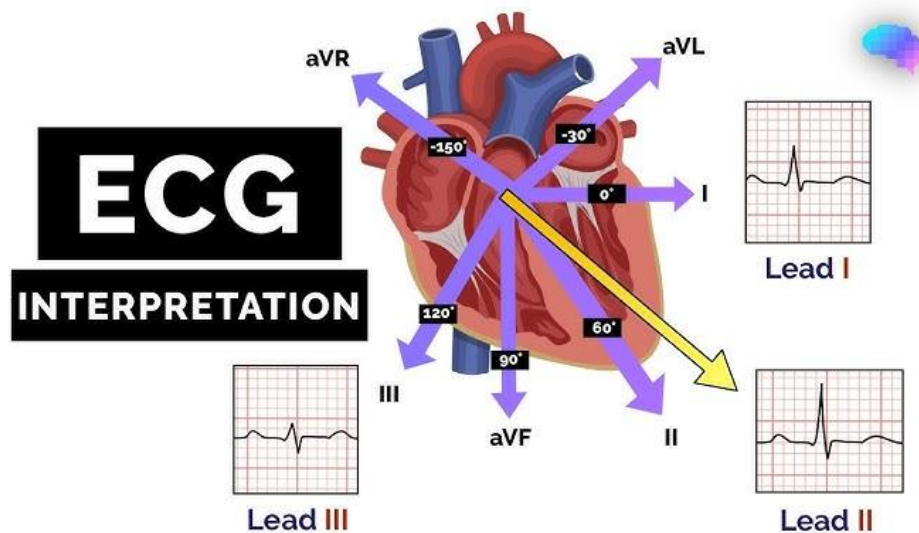
- 4 điện cực chi: Gắn tại hai cổ tay và hai cổ chân.
- 6 điện cực ngực: Gắn ở các vị trí V1 đến V6 quanh vùng ngực.



Hình 2: Vị trí đo của các chuyển đạo

- Các chuyển đạo được chia thành 3 nhóm:
 - Chuyển đạo chi (limb leads): Chuyển đạo cực chuẩn: I, II, III
 - Chuyển đạo tăng cường: aVR, aVL, aVF
 - Chuyển đạo ngực (precordial leads): V1 đến V6

Góc nhìn tim: Mỗi chuyển đạo quan sát tim từ một góc độ khác nhau, tạo nên một “hình ảnh điện học ba chiều” của tim:



Hình 3: Minh họa đo chuyển đạo I, II, III

- V1–V2: vách tim (septal)
- V3–V4: thành trước (anterior)
- V5–V6: thành bên trái (lateral)

- II, III, aVF: thành dưới (inferior)
- I, aVL: thành bên cao (high lateral)
- aVR: nhìn từ phải trên (ít dùng để chẩn đoán chính nhưng có giá trị định hướng)

1.2.2. Cơ chế hoạt động

Khi tim co bóp, dòng điện nội tại lan truyền qua cơ tim, tạo ra hiệu điện thế giữa các vị trí trên cơ thể. Các điện cực ghi lại sự thay đổi điện áp này theo thời gian, tạo thành sóng ECG.

Tín hiệu ECG thường được ghi lại trong khoảng thời gian 10 giây với tần số lấy mẫu 500–1000 Hz trên giây, cho phép phân tích chính xác các biến đổi nhỏ trong chu kỳ tim.

1.3. Cấu trúc sóng điện tâm đồ và ý nghĩa lâm sàng

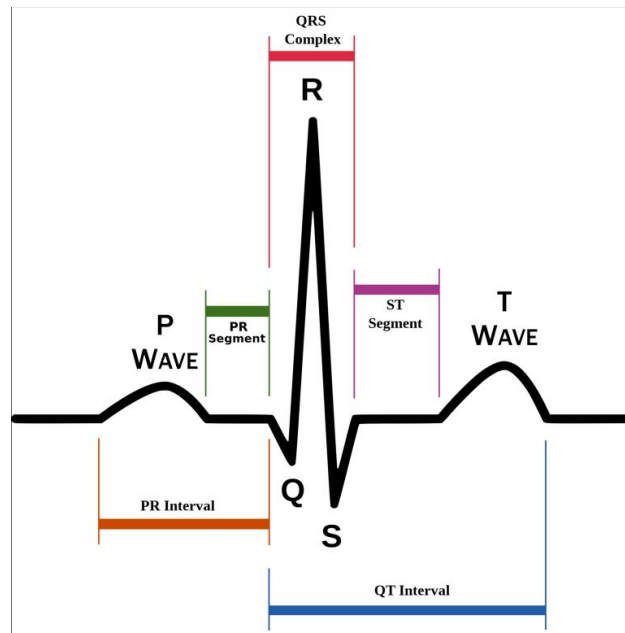
Một chu kỳ tim điển hình trên ECG bao gồm các thành phần chính:

- Sóng P: quá trình khử cực liên tiếp của tâm nhĩ phải và trái
- Khoảng PR: khoảng thời gian từ lúc bắt đầu khử cực tâm nhĩ (sóng P) đến lúc bắt đầu khử cực thất (phức bộ QRS)
- Phức hợp QRS: khử cực tâm thất phải và trái

Sóng Q: sóng âm đầu tiên của phức bộ

Sóng R: là sóng dương đầu tiên, theo sau nó là sóng âm S

- Đoạn ST: thể hiện rằng quá trình khử cực cơ tim tâm thất đã hoàn thành (kết thúc phức bộ QRS (điểm J) → đến bắt đầu sóng T)
- Sóng T: tái cực thất
- Khoảng QT: thời gian toàn bộ chu kỳ khử cực và tái cực thất (từ khi bắt đầu khử cực tâm thất cho đến khi kết thúc tái cực tâm thất)



Hình 4: Nhận biết các sóng P, PR, ST, QRS, QT, T

1.4. Ứng dụng của ECG 12 chuyển đạo trong chẩn đoán bệnh lý tim mạch

ECG 12-lead là công cụ chủ lực trong chẩn đoán nhiều bệnh lý tim mạch, đặc biệt là các rối loạn về nhịp tim và dẫn truyền. Trong khuôn khổ đề tài này, chúng tôi lựa chọn 8 nhãn bệnh phổ biến nhất trong tập dữ liệu PhysioNet - ECG Arrhythmia và có giá trị lâm sàng cao, chiếm tỷ lệ lớn và đại diện cho nhiều dạng bất thường khác nhau của tim. Các nhãn bao gồm:

Mã nhãn	Tên bệnh lý	Tên tiếng Anh	Tỷ lệ xuất hiện (%)
SB	Nhịp chậm xoang	Sinus Bradycardia	23.4
SR	Nhịp xoang bình thường	Sinus Rhythm	11.3
AF	Cuồng nhĩ	Atrial Flutter	11.3
ST	Chênh lên đoạn ST	ST Elevation	10.3
SA	Loạn nhịp xoang	Sinus Arrhythmia	3.6
AFIB	Rung nhĩ	Atrial Fibrillation	2.5
STDD	Chênh xuống	ST Depression	2.35

	đoạn ST		
ALS	Trục điện tim lệch trái	Left Axis Deviation	2.17

Bảng 1: Bảng mã dùng để nhận diện

1.5. Đặc điểm ECG theo chuyển đạo của 8 nhân bệnh

Mỗi rối loạn tim mạch thường biểu hiện đặc trưng ở một số chuyển đạo nhất định trên điện tâm đồ. Ta có thước đo của ECG:

- Tốc độ giấy: 50 mm/s (tần số lấy mẫu 500Hz/s)
- ➔ 1 ô nhỏ = 0,02 giây, 1 ô lớn = 0,1 giây
- ➔ Trong 1 giây có 50 mm = 50 ô nhỏ = 10 ô lớn
- 1 phút = 60 giây ➔ 600 ô lớn

$$HR = \frac{60}{T_{RR}} = \frac{60}{N_{small} \times 0.02} = \frac{3000}{N_{small}} \quad (\text{bpm})$$

Hình 5: Công thức vẽ lưới ECG

Ví dụ: 60 ô nhỏ ➔ $HR = 3000/60 = 50 \text{ bpm}$

1.5.1. SR – Nhịp xoang bình thường

- Tần số tim: 60 – 100 bpm (50 – 30 ô nhỏ giữa hai đỉnh R)
- Sóng P bình thường, đều, giống nhau, mỗi P đều dẫn xuống QRS
- Sóng P dương ở I, II, aVF; âm ở aVR
- Khoảng PR: 0,12 – 0,2 giây tương đương 6-10 ô nhỏ
- Phức bộ QRS: hẹp, bình thường: $\leq 0,1$ giây.
- Khoảng RR: đều nhau ➔ nhịp đều

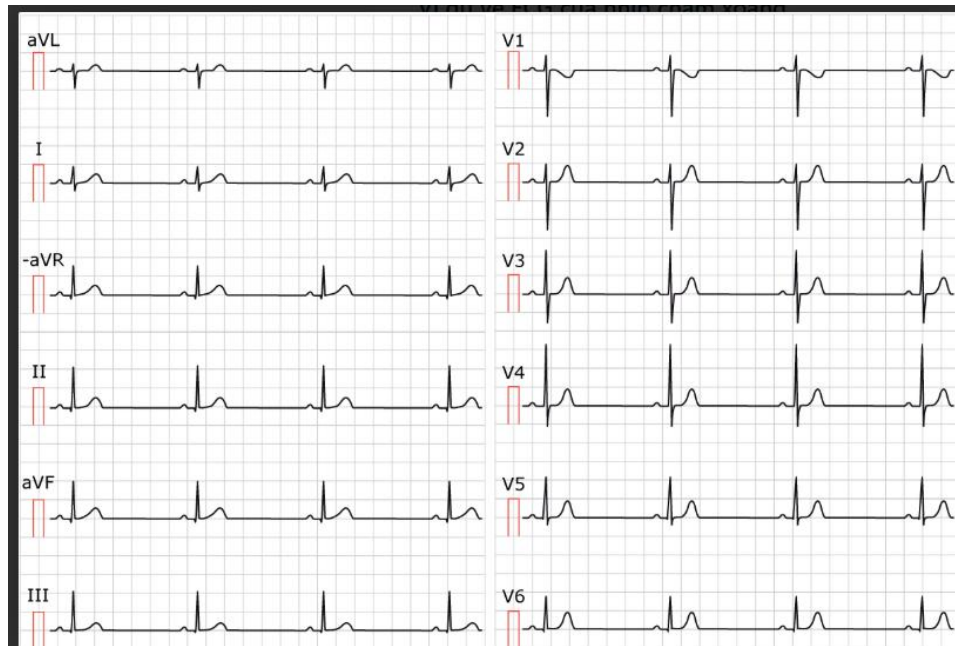
1.5.2. SB – Nhịp chậm xoang

Nhịp chậm xoang đáp ứng các tiêu chuẩn của nhịp xoang nhưng nhịp tim chậm hơn 50 nhịp/phút (60 ô nhỏ), nặng khi dưới 40 bpm (trên 75 ô nhỏ).

Các tiêu chuẩn điện tâm đồ như sau:

- Nhịp tim đều đặn với nhịp tim thất chậm hơn 50 nhịp mỗi phút
- Sóng P có hình thái không đổi trước mỗi phức hợp QRS

- Sóng P dương ở chuyển đạo chi II



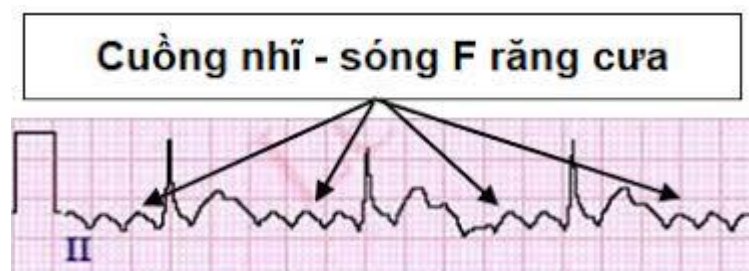
Hình 6: Minh họa lưới ECG của Nhịp chậm xoang

Tốc độ giấy của hình trên là 25 mm/giây

Tính tần số bằng cách lấy 300 chia cho số ô giữa hai chu kỳ (ví dụ: giữa hai sóng R). Như hình minh họa, có khoảng 6.5 ô giữa hai sóng R $\rightarrow 300/6.5$ bằng 46 nhịp/phút.

1.5.3. AF – Cuồng nhĩ

- Sóng P: Không có P riêng biệt, thay bằng sóng răng cưa (F waves) đều đặn, tần số
- Hình dạng sóng F: giống răng cưa, rõ nhất ở lead II, III, aVF, chuỗi F nối nhau liên tục, cách nhau chưa tới 1 ô lớn, gần như không có đường đẳng điện giữa chúng.
- RR có thể đều



Hình 7: Minh họa sóng bệnh Cuồng nhĩ

1.5.4. ST - ST Chênh lệch

- Bình thường, đoạn ST nằm ngang đường đẳng điện (baseline), nghĩa là ngang với đoạn PR → nhìn như một đường thẳng giữa QRS và T.
- ST chênh lệch là khi đoạn ST bị đẩy lên cao hơn baseline → tim đang bị tổn thương, thiếu máu nghiêm trọng (nhồi máu cơ tim cấp).



Hình 8: Minh họa sóng ST chênh lệch

- Tiêu chuẩn ST chênh lệch theo ESC/AHA

Đo tại điểm J – điểm kết thúc của phức bộ QRS, ta sẽ đo độ cao của đoạn ST tại điểm J so với baseline, như hình trên là 2.5 mm.

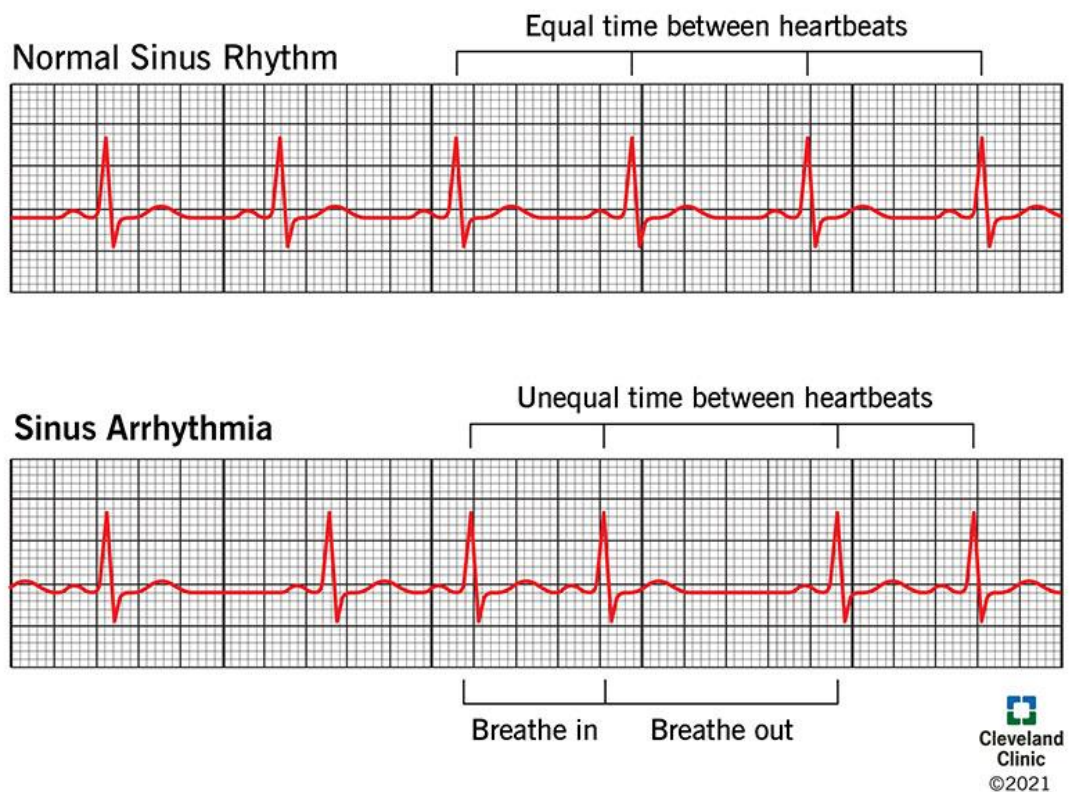
Lead	Giới tính	Mức chênh lệch
V2-V3	Nam < 40 tuổi	$\geq 2.5 \text{ mm (0.25mV)}$
	Nam ≥ 40 tuổi	$\geq 2 \text{ mm}$
	Nữ	$\geq 1.5 \text{ mm}$
Các lead còn lại	Nam và Nữ	$\geq 1 \text{ mm}$

Bảng 2: Phân biệt mức chênh lệch độ đo theo giới tính cho các lead

1.5.5. SA – Loạn nhịp xoang

- Khoảng cách giữa các nhịp tim (R-R) không đều nhau.
- Tuy nhiên, dạng sóng P, QRS, T vẫn bình thường → tim vẫn theo nhịp xoang, chỉ là tốc độ nhanh – chậm thay đổi một cách chu kỳ.
- Cách nhận biết: đo khoảng RR liên tiếp: nếu chênh lệch trên 0,12 giây (3 ô nhỏ) → loạn nhịp xoang
- Khoảng cách giữa các nhịp tim (R-R) không đều nhau.
- Tuy nhiên, dạng sóng P, QRS, T vẫn bình thường → tim vẫn theo nhịp xoang, chỉ là tốc độ nhanh – chậm thay đổi một cách chu kỳ.

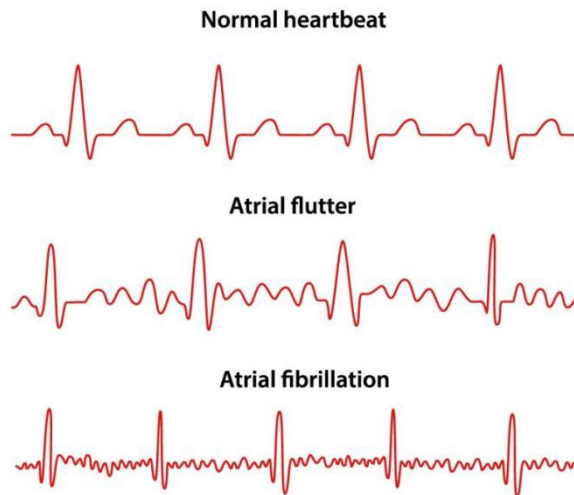
- Cách nhận biết: đo khoảng RR liên tiếp: nếu chênh lệch trên 0,12 giây (3 ô nhỏ) → loạn nhịp xoang



Hình 9: So sánh giữa nhịp xoang bình thường và nhịp chậm xoang

1.5.6. AFIB – Rung nhĩ

- Không có sóng P thật, thay bằng sóng f nhỏ, lẫn tăn, không đều, biên độ thấp, tần số cao (350-600 lần/phút tương đương 8.5 và 5 ô nhỏ)
- Khoảng R-R hoàn toàn không đều, không có chu kỳ lặp lại
- Phức bộ QRS: Hình dạng bình thường
- Tần số thất: thường nhanh, 100-160 bpm (30 và 19 ô nhỏ)



Hình 10: So sánh sóng của người bình thường, cuồng nhĩ và rung nhĩ

1.5.7. STDD – ST chênh xuống

- Đoạn ST nằm thấp hơn đường đẳng điện ($\geq 0.5 - 1\text{mm}$)
- Từ J-point, đo cao độ của đoạn ST so với baseline (đoạn PR).
- Nếu ST thấp hơn baseline $\geq 0.5-1\text{ mm} \rightarrow$ ST chênh xuống.



Hình 11: Minh họa sóng ST chênh xuống

- Phải có ít nhất 2 leads liên tiếp cùng vùng bị thay đổi mới có giá trị chẩn đoán
- Các lead trên ECG được chia theo vùng như sau:

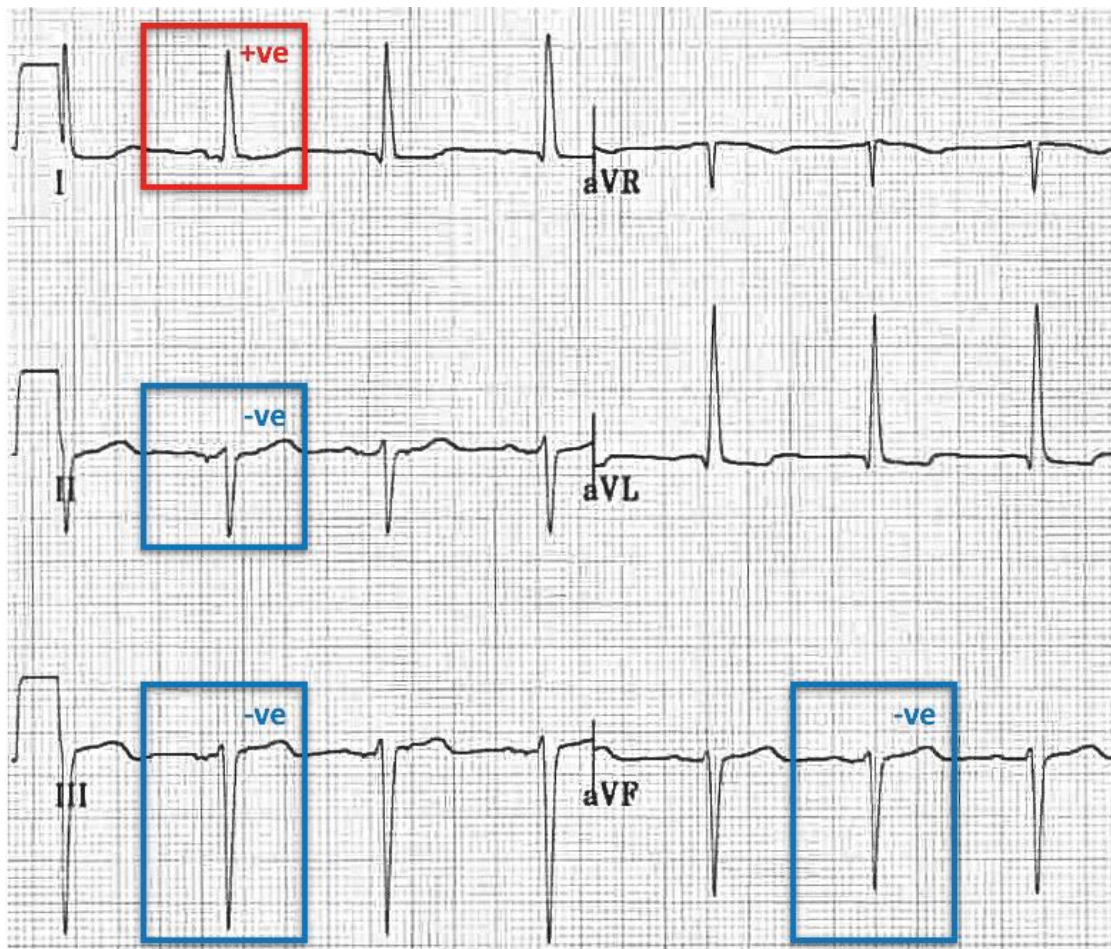
Vùng tim	Lead cùng vùng	Giải thích
Thành dưới (Inferior)	II – III - aVF	Nhìn vùng dưới của thất trái
Thành bên (Lateral)	I – aVL – V5 -V6	Nhìn vùng bên trái của tim

Thành trước (Anterior)	V1- V4	Nhìn vùng trước và vách liên thất
Thành trước – bên (Anterolateral)	V3-V6 + I - aVL	Kết hợp vùng trước và bên
Thành sau (Posterior)	Không có lead trực tiếp, nhưng V1- V3 phản chiếu → Thấy ST chênh xuống ở đây gợi ý nhồi máu thành sau	

Bảng 3: Chia lead theo vùng

1.5.8. ALS – Trục điện tim lệch trái

- Dấu hiệu nhận biết: QRS dương ở lead I, âm ở lead aVF



Hình 12: Minh họa sóng trục điện tim lệch trái

- Sau khi xác định lệch trái, dùng lead II để xác định mức độ lệch trái có sâu không:

Lead II	Kết luận
Vấn dương (R lớn hơn S)	Lệch trái nhẹ, có thể vẫn bình thường ở người già
Âm (R nhỏ, S lớn)	Lệch trái thật sự (ALS rõ rệt)

Bảng 4: Dùng lead II để xác định độ lệch trái

Tín hiệu ECG là đa chiều và phức tạp, do đó việc phân tích không thể chỉ dựa trên một chuyển đạo đơn lẻ, mà cần phối hợp toàn bộ 12 chuyển đạo để đưa ra kết luận chính xác. Đây cũng chính là cơ sở quan trọng để ứng dụng mô hình học sâu có khả năng học đồng thời các mối tương quan giữa nhiều kênh tín hiệu.

1.6. Cơ sở cho việc áp dụng học sâu và giải thích mô hình trên ECG

Nhờ đặc điểm có cấu trúc, tính tuần hoàn và hình thái đặc trưng, tín hiệu ECG rất phù hợp với các kỹ thuật học sâu, đặc biệt là mạng nơ-ron tích chập một chiều (1D-CNN), vốn mạnh trong việc học các đặc trưng cục bộ trên chuỗi thời gian.

Tuy nhiên, vì tính chất nhạy cảm trong ứng dụng y tế, các mô hình học sâu cần không chỉ chính xác, mà còn phải giải thích được quyết định của mình. Do đó, việc tích hợp các kỹ thuật như Grad-CAM để trực quan hóa vùng tín hiệu có ảnh hưởng lớn đến dự đoán là cực kỳ cần thiết để tạo niềm tin cho bác sĩ lâm sàng và hỗ trợ quá trình ra quyết định y khoa.

Từ các đặc điểm sóng ECG cụ thể theo từng chuyển đạo, có thể thấy tiềm năng lớn trong việc áp dụng mô hình học sâu để tự động hóa quá trình chẩn đoán. Việc tận dụng toàn bộ 12 chuyển đạo đồng thời kết hợp với phương pháp trực quan hóa như Grad-CAM sẽ giúp nâng cao hiệu quả, tính chính xác, và mức độ tin cậy của hệ thống hỗ trợ chẩn đoán bằng AI – là mục tiêu trung tâm của đề tài này.

Chương 2: Mô hình ResNet và Inception Time

2.1. Tổng quan về học sâu trong phân tích tín hiệu ECG

Trong những năm gần đây, các mô hình học sâu (deep learning) đã chứng minh hiệu quả vượt trội trong việc xử lý các dạng dữ liệu phức tạp, bao gồm hình ảnh, âm thanh và chuỗi thời gian. Đặc biệt trong lĩnh vực y học, học sâu ngày càng được quan tâm nhờ khả năng tự động học đặc trưng và phát hiện các bất thường trong dữ liệu sinh lý như điện tâm đồ (ECG).

Với tính chất là một tín hiệu sinh học liên tục, nhiều chiều và mang đặc điểm cục bộ rõ rệt, ECG rất phù hợp để xử lý bằng các mô hình mạng nơ-ron tích chập một chiều (1D Convolutional Neural Networks – 1D-CNN). Các mô hình này có khả năng tự động trích xuất các đặc trưng quan trọng trên chuỗi tín hiệu theo thời gian mà không cần tiền xử lý phức tạp.

Tuy nhiên, một hạn chế lớn của các mạng CNN truyền thống là khi độ sâu tăng lên, mô hình dễ gặp phải hiện tượng suy giảm gradient (vanishing gradient), dẫn đến mất thông tin và hiệu suất kém. Để giải quyết vấn đề này, kiến trúc Residual Network (ResNet) đã được giới thiệu như một giải pháp hiệu quả, cho phép xây dựng các mô hình học sâu với hàng chục, thậm chí hàng trăm lớp mà vẫn đảm bảo hiệu quả huấn luyện.

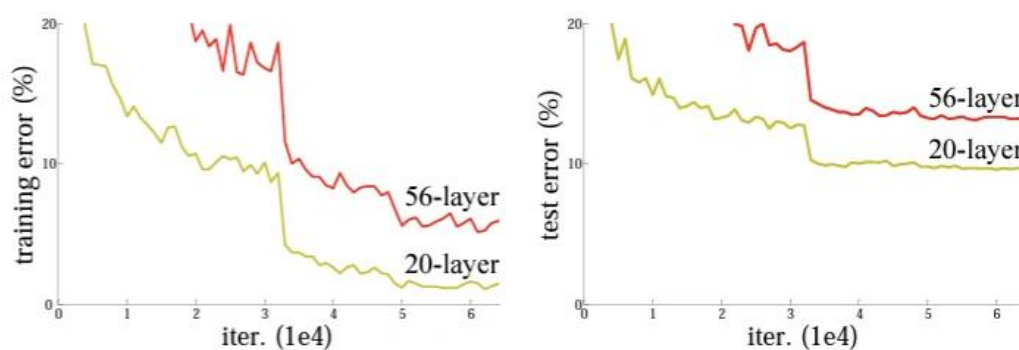
2.2. Kiến trúc ResNet và phiên bản ResNet1D-CNN

2.2.1. Mạng ResNet là gì?

Mạng ResNet hay còn được gọi là mạng nơ-ron dư thừa, là một kiến trúc học sâu mà trong đó các lớp học các hàm dư (residual functions) liên quan đến đầu vào của lớp. Được cho ra mắt và phát triển bởi một nhóm nghiên cứu thuộc Microsoft gồm Kaiming He, Xiangyu Zhang, Shaoqing Ren và Jian Sun với mục đích nhằm để nhận diện hình ảnh và giành chiến thắng trong cuộc thi ILSVRC năm 2015.

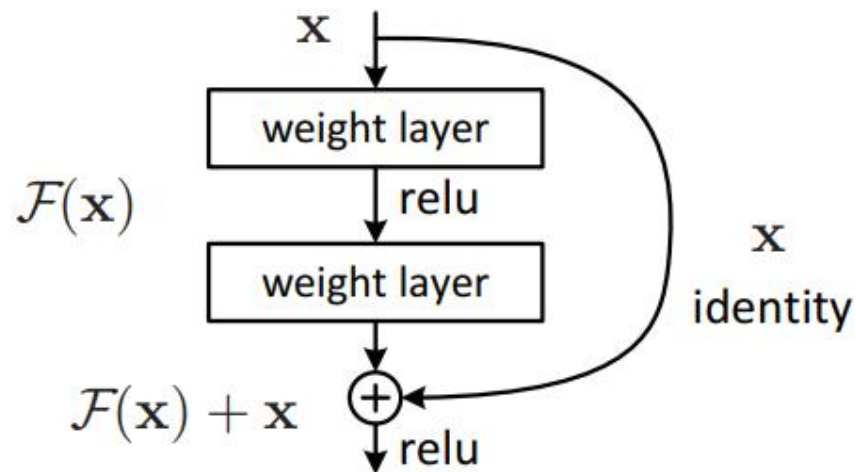
Trước khi ResNet được nghiên cứu và cho ra đời, cũng đã có những mô hình khác xuất hiện như: Plain Network, LeNet hay VGG. Tuy nhiên, những mô hình này xuất hiện vấn đề về Vanishing Gradient (Tiêu Biến Độ Dốc) hay Exploding Gradient (Bùng Nổ Độ Dốc), là hiện tượng khi mà mô

hình được huấn luyện và học tập từ dữ liệu, khi tính toán hàm mất mát (loss function) trong quá trình lan truyền ngược (backpropagation) độ dốc (gradient) của hàm quá nhỏ hoặc quá lớn khiến cho các trọng số không được cập nhật hay trọng số quá lớn làm cho mô hình huấn luyện trở nên chậm đi hoặc không hội tụ được khiến hiệu suất của mô hình giảm đi rất nhiều, với đầu vào là dữ liệu mới có thể dẫn tới việc dự đoán không chính xác gây ra sai lệch lớn. Có thể thấy được thông qua ảnh dưới, khi huấn luyện dữ liệu ta tăng số lớp lên từ 20 lên 56 lớp thì tỉ lệ lỗi khi huấn luyện lại tăng cao với mô hình dùng 56 lớp, lẽ ra với số lớp càng nhiều thì trọng số phải càng được tối ưu và cho ra kết quả tốt hơn mô hình có số lớp nhỏ, ấy vậy mà chúng lại đi ngược chứng tỏ rằng mô hình giảm hiệu quả rất nhiều do 2 vấn đề ta đã đề cập trước đó.



Hình 13: Tỷ lệ training error

Mạng ResNet ra đời nhóm nghiên cứu đã đưa ra khái niệm “Khối dư” (residual block) và sử dụng kỹ thuật Skip Connection (bỏ qua kết nối) mục đích là để bỏ qua kết nối các hoạt động của một lớp với các lớp tiếp theo bằng việc bỏ qua một số lớp ở giữa, điều này tạo thành Khối Dư dẫn đến các mạng được tạo ra bằng cách xếp chồng các khối dư này lại tạo thành kiến trúc mạng nơ-ron dư thừa. Việc đưa ra khái niệm mới và kỹ thuật Skip Connection nhằm để giải quyết vấn đề về hiện tượng Tiêu Biên/Bùng Nổ độ dốc - vấn đề mà những mô hình trước đó chưa làm được.



Hình 14: Kỹ thuật Skip Connection

Với:

- x là tập hợp các lớp tích chập
- $F(x)$ là hàm phần dư (residual function)
- $H(x)$ là hàm ánh xạ từ đầu vào đến đầu ra

Thay vì học một ánh xạ trực tiếp từ $H(x)$ như những mạng khác, mạng ResNet sẽ học một ánh xạ phần dư $F(x) = H(x) - x$ và đầu ra cuối cùng là $F(x) + x$. Điều này giúp cải thiện và hạn chế được những vấn đề về Tiêu Biến Độ Dốc và Bùng Nổ Độ Dốc.

2.2.2. ResNet1D-CNN cho tín hiệu ECG

ResNet1D là phiên bản mở rộng của ResNet gốc, được điều chỉnh để làm việc với dữ liệu một chiều như tín hiệu ECG. Thay vì sử dụng các lớp tích chập hai chiều như trong xử lý ảnh, ResNet1D sử dụng Conv1D để học đặc trưng từ chuỗi thời gian.

Khi áp dụng cho tín hiệu ECG 12 chuyển đạo, ResNet1D-CNN có một số lợi thế nổi bật:

- Trích xuất đặc trưng cục bộ hiệu quả từ các vùng có sóng P, QRS, ST trong từng chuyển đạo

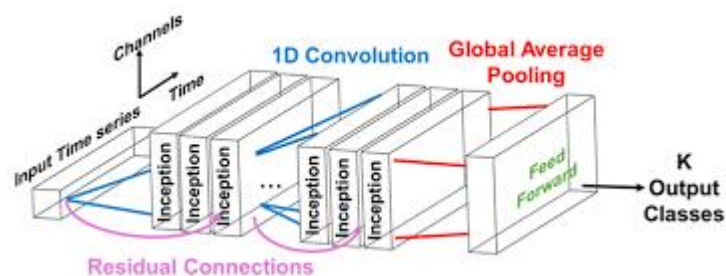
- Học được mối quan hệ liên kênh giữa các chuyển đạo – rất quan trọng trong việc chẩn đoán các bệnh lý như nhồi máu cơ tim, rối loạn nhịp
- Khả năng mở rộng kiến trúc sâu mà vẫn đảm bảo tính ổn định trong huấn luyện

Với những ưu điểm trên, ResNet1D-CNN đã được lựa chọn làm mô hình học sâu chủ đạo trong đề tài nhằm giải quyết bài toán phân loại 8 nhãn bệnh tim thông qua tín hiệu ECG.

2.3. InceptionTime cho tín hiệu ECG

InceptionTime là một kiến trúc mạng CNN 1D chuyên dụng cho các chuỗi thời gian (time series). Mạng này tận dụng các Inception module với nhiều kích thước kernel khác nhau để học các đặc trưng ở nhiều độ dài, đồng thời sử dụng residual connection để hỗ trợ huấn luyện sâu mà không bị mất gradient.

Khi áp dụng cho tín hiệu ECG, InceptionTime có thể xử lý trực tiếp dữ liệu đa kênh (multi-lead), ví dụ 12-lead ECG, mà không cần chuyển đổi sang dạng 2D. Kiến trúc này đặc biệt phù hợp với ECG vì nó cho phép mạng học được cả pattern ngắn hạn (P wave, QRS complex) và pattern dài hạn (ST segment, RR interval) cùng lúc. Ngoài ra, mạng có thể kết hợp cơ chế attention để tập trung vào những đoạn tín hiệu quan trọng, giúp cải thiện hiệu quả phân loại.



Hình 15: Kiến trúc mô hình InceptionTime

Cấu trúc cơ bản của InceptionTime cho ECG gồm:

- Input: tensor [batch_size, n_channels, seq_len] với n_channels là số lead.
- Chuỗi InceptionResidualBlock 1D: mỗi block bao gồm nhiều convolution 1D với kernel size khác nhau kết hợp residual connection.
- Global Average Pooling (GAP): tổng hợp thông tin dọc theo trục thời gian, tạo vector đặc trưng cố định.
- Fully Connected Layer: đưa vector đặc trưng vào lớp phân loại cuối cùng để dự đoán nhãn.

Ưu điểm của InceptionTime với ECG:

- Giữ nguyên thông tin chi tiết từng lead.
- Bắt được pattern ngắn hạn lẫn dài hạn trong tín hiệu.
- Hỗ trợ huấn luyện sâu nhờ residual connection.
- Có thể mở rộng thêm attention hoặc ensemble để nâng cao độ chính xác phân loại.

2.4. Giải thích mô hình bằng Attention

Một trong những hạn chế lớn của các mô hình học sâu là tính "hộp đen" – khó giải thích tại sao mô hình lại đưa ra một quyết định cụ thể. Điều này đặc biệt quan trọng trong lĩnh vực y tế, nơi mà tính minh bạch và khả năng kiểm chứng là yêu cầu thiết yếu để có thể ứng dụng thực tế.

Để giải quyết vấn đề này, đề tài sử dụng cơ chế attention nhằm xác định các vùng tín hiệu mà mô hình tập trung khi dự đoán.

Nguyên lý:

- Attention học trọng số trên trục thời gian, biểu thị mức độ “quan trọng” của từng điểm tín hiệu đối với dự đoán cuối cùng.
- Các giá trị attention thường được chuẩn hóa về khoảng [0, 1], giúp so sánh mức độ quan trọng giữa các vùng của ECG.

Cách đánh giá trực quan:

- Biểu diễn tín hiệu ECG 12-lead trên lưới chuẩn (50 mm/s, 10 mm/mV).
- Chấm các điểm có attention cao lên sóng ECG, thường dùng màu đỏ hoặc nổi bật để nhận biết vùng mạng “chú ý”.

- Việc này giúp:
 - Hiểu rõ vùng ECG ảnh hưởng nhiều nhất đến dự đoán.
 - Đánh giá tính hợp lý của mô hình, ví dụ mạng có chú ý đến QRS complex hoặc ST segment đúng như kiến thức lâm sàng.
 - Minh bạch mô hình (explainable AI) khi áp dụng cho chẩn đoán y tế.

Ý nghĩa đối với ECG:

- Giúp xác định liệu mô hình có dựa vào những đặc trưng sinh lý quan trọng hay không.
- Có thể hỗ trợ bác sĩ hoặc nhà nghiên cứu kiểm tra, validate các dự đoán của mô hình.
- Khi attention tập trung vào các đoạn không liên quan, có thể gợi ý điều chỉnh mô hình hoặc dữ liệu huấn luyện.

Chương 3: Thu thập và tiền xử lý dữ liệu

3.1. Thu thập dữ liệu

Trong nghiên cứu này, chúng em sử dụng bộ dữ liệu thu tín hiệu từ máy đo điện tâm đồ 12 chuyển đạo do các tác giả Jianwei Zheng, Hangyuan Guo, Huimin Chu đăng tải trên trang PhysioNet với tựa đề “A large scale 12-lead electrocardiogram database for arrhythmia study” (Cơ sở dữ liệu điện tâm đồ 12 chuyển đạo quy mô lớn phục vụ nghiên cứu rối loạn nhịp tim) cho mục đích nghiên cứu.

Bộ dữ liệu gồm có:

- WFDBRecords: lưu trữ tín hiệu điện tâm đồ 12 chuyển đạo của 45152 bệnh nhân
- ConditionNames_SNOMED-CT: lưu trữ 64 mã bệnh (Snomed CT)
- RECORDS: lưu trữ đường dẫn đến các thư mục bệnh nhân

Name	Size	Modified
WFDBRecords		
ConditionNames_SNOMED-CT.csv	2.5 KB	2022-02-14
LICENSE.txt	14.5 KB	2022-08-10
RECORDS	8.8 KB	2022-07-26
SHA256SUMS.txt	8.3 MB	2022-08-24

Hình 16: Cấu trúc thư mục dữ liệu huấn luyện

Nguồn dữ liệu:

<https://physionet.org/content/ecg-arrhythmia/1.0.0/>

3.2. Tiền xử lý dữ liệu

Sau khi tải về bộ dữ liệu và giải nén, thu được dữ liệu ban đầu với tổng là 5.1 GB. Để tiến hành quá trình huấn luyện dữ liệu, trước hết ta phải lọc được mã id của các bệnh nhân, tuổi, giới tính, đường dẫn đến file dữ liệu của bệnh nhân và mã bệnh của họ rồi lưu lại thành file dữ liệu bệnh nhân.

record_id	file_path	age	sex	diagnosis_code	diagnosis_name
JS00001	E:\NCKH - 2026\ECG 12leads\ECG\WFDBRecords\01\010\JS00001.he	85	Male	164889003,5911800	AFIB,RBBB,TWC
JS00002	E:\NCKH - 2026\ECG 12leads\ECG\WFDBRecords\01\010\JS00002.he	59	Female	426177001,1649340	SB,TWC
JS00004	E:\NCKH - 2026\ECG 12leads\ECG\WFDBRecords\01\010\JS00004.he	66	Male	426177001	SB
JS00005	E:\NCKH - 2026\ECG 12leads\ECG\WFDBRecords\01\010\JS00005.he	73	Female	164890007,4296220	AF,STDD,STTC
JS00006	E:\NCKH - 2026\ECG 12leads\ECG\WFDBRecords\01\010\JS00006.he	46	Female	426177001	SB
JS00007	E:\NCKH - 2026\ECG 12leads\ECG\WFDBRecords\01\010\JS00007.he	80	Female	164889003,1649340	AFIB,TWC
JS00008	E:\NCKH - 2026\ECG 12leads\ECG\WFDBRecords\01\010\JS00008.he	46	Male	426783006	SR
JS00009	E:\NCKH - 2026\ECG 12leads\ECG\WFDBRecords\01\010\JS00009.he	45	Male	426177001	SB
JS00010	E:\NCKH - 2026\ECG 12leads\ECG\WFDBRecords\01\010\JS00010.he	47	Female	426177001	SB
JS00011	E:\NCKH - 2026\ECG 12leads\ECG\WFDBRecords\01\010\JS00011.he	63	Male	426177001	SB
JS00012	E:\NCKH - 2026\ECG 12leads\ECG\WFDBRecords\01\010\JS00012.he	77	Male	427084000	ST
JS00013	E:\NCKH - 2026\ECG 12leads\ECG\WFDBRecords\01\010\JS00013.he	54	Male	426177001,2704920	SB,1AVB
JS00014	E:\NCKH - 2026\ECG 12leads\ECG\WFDBRecords\01\010\JS00014.he	74	Male	426177001,2511730	SB,ABI
JS00015	E:\NCKH - 2026\ECG 12leads\ECG\WFDBRecords\01\010\JS00015.he	54	Female	426177001	SB
JS00016	E:\NCKH - 2026\ECG 12leads\ECG\WFDBRecords\01\010\JS00016.he	67	Female	164889003,4287500	AFIB,STTC
JS00017	E:\NCKH - 2026\ECG 12leads\ECG\WFDBRecords\01\010\JS00017.he	51	Female	426177001	SB
JS00018	E:\NCKH - 2026\ECG 12leads\ECG\WFDBRecords\01\010\JS00018.he	61	Male	426177001,5911800	SB,RBBB
JS00019	E:\NCKH - 2026\ECG 12leads\ECG\WFDBRecords\01\010\JS00019.he	68	Female	426177001,2704920	SB,1AVB,TWO
JS00020	E:\NCKH - 2026\ECG 12leads\ECG\WFDBRecords\01\010\JS00020.he	70	Female	426177001	SB

Hình 17: Nội dung file dữ liệu bệnh nhân

Trong quá trình xử lý dữ liệu, sẽ có một số bệnh nhân có giới tính unknown, tuổi bị null, do số lượng ít trên tổng hơn 45 nghìn dữ liệu nên ta loại bỏ đi các bệnh nhân này, kết quả cuối cùng thu lại được còn lại 44203 bệnh nhân. Ngoài ra, để xây dựng mô hình nhận diện dữ liệu và dự đoán đa nhãn, ta cần phải truy xuất ra những nhãn quan trọng, xuất hiện chủ chốt trong bộ dữ liệu. Do các mã bệnh vẫn còn tồn tại dưới dạng mã viết tắt chuyên sử dụng trong y khoa, nên ta sẽ đối chiếu với từ điển, lọc tự động để tìm ra tên gốc của mã bệnh, từ đó lưu lại thành file nhãn bệnh hoàn chỉnh:

Snomed_CT	Resolved_Name
81898007	Ventricular escape rhythm
164931005	ST elevation
164909002	ECG: left bundle branch block
164912004	EKG: P wave abnormal
426995002	EKG: junctional escape rhythm
284470004	Premature atrial contraction
418818005	Brugada syndrome
251198002	Clockwise cardiac rotation
233897008	Re-entrant atrioventricular tachycardia
713422000	EKG: atrial tachycardia
233892002	Ectopic atrial tachycardia
75532003	Ventricular escape beat
57054005	Acute myocardial infarction
426648003	ECG: junctional tachycardia
251173003	Atrial bigeminy
6.7741E+13	Left atrial enlargement
74390002	Wolff-Parkinson-White pattern
54016002	Mobitz type I incomplete atrioventricular block
426177001	ECG: sinus bradycardia
164942001	ECG: F wave present
17366009	Atrial arrhythmia

Hình 18: Nội dung file chứa dữ liệu mã bệnh đã được mapping

Sau khi đã có tên gốc của nhãn bệnh, ta tiến hành lọc ra những nhãn bệnh xuất hiện phổ biến, thu được top 10 nhãn sau:

	Code	Abbreviation	Full_Name	Count	Percentage
0	426177001	SB	Sinus Bradycardia	16542	23.429267
1	164890007	AF	Atrial Flutter	8010	11.344966
2	426783006	SR	Sinus Rhythm	7962	11.276981
3	427084000	ST	ST Elevation	7244	10.260042
4	164934002	TWC	T-wave Changes	7006	9.922951
5	59931005	TWO	T Wave Inversion	2866	4.059260
6	427393009	SA	Sinus Arrhythmia	2546	3.606028
7	164889003	AFIB	Atrial Fibrillation (alternate)	1780	2.521104
8	429622005	STDD	ST drop down	1665	2.358223
9	39732003	ALS	Axis left shift	1537	2.176930

Hình 19: Top 10 nhãn bệnh phổ biến

Tiếp tục lựa chọn các nhãn để dự đoán từ 10 nhãn trên loại bỏ đi 2 nhãn không có tính lâm sàng cao trong chẩn đoán bệnh lần lượt là TWC và TWO (Biến đổi sóng T và đảo ngược sóng T), sau cùng ta còn lại 8 nhãn. Từ đây tiến hành triển khai 8 nhãn này để huấn luyện dữ liệu về sau.

Các bệnh nhân được lưu trữ thông tin qua 2 files: .hea và .mat. Đối với file .hea lưu trữ thông số của 12 chuyển đạo điện tâm đồ, tuổi, giới tính và mã bệnh của

bệnh nhân. Đối với file .mat thì lưu trữ tín hiệu đo được từ điện tâm đồ. Việc lọc dữ liệu bệnh nhân, tên, tuổi, giới tính và đường dẫn đã được ta xử lý thông qua file .hea, việc tiếp theo cần làm chính là xử lý file tín hiệu .mat.

Trong quá trình đo điện tim, thường tín hiệu điện tim đo được sẽ “không hoàn hảo” vì bị nhiễu bởi những tạp âm bên ngoài khác như tiếng ma sát khi cử động cơ thể, tiếng điện lưới. Nếu những âm thanh này không được xử lý, chúng sẽ xen lẫn với tín hiệu thực sự đo được và làm nhiễu, từ đó nếu sử dụng tín hiệu nguyên bản chưa qua lọc để huấn luyện dễ dẫn đến việc dự đoán sai lệch, thiếu tính chính xác. Trong dữ liệu gốc, thông tin lưu ở file .hea cho thấy tần số lấy mẫu của dữ liệu là 500Hz/s, độ nhạy (gain) 1000/mV và đường đẳng điện (baseline) đặt ở mức 0.

Vì vậy ta sẽ tiến hành lọc nhiễu như sau cho toàn bộ 12 chuyển đạo:

- Lọc nhiễu điện lưới ở tần số 50Hz
- Lọc nhiễu và giữ lại tần số chính cho việc nhận diện điện tâm đồ (0.5 - 40Hz)
- Chuẩn hóa tín hiệu raw về dạng mV

Lọc nhiễu điện:

```
def apply_notch(sig, fs, notch_freq=50.0, quality=30.0):
    """Loại nhiễu điện 50Hz (VN/EU powerline)"""
    b, a = iirnotch(notch_freq / (fs / 2), quality)
    return filtfilt(b, a, sig)
```

Lọc tần số, giữ lại tín hiệu trong khoảng 0.5-40Hz:

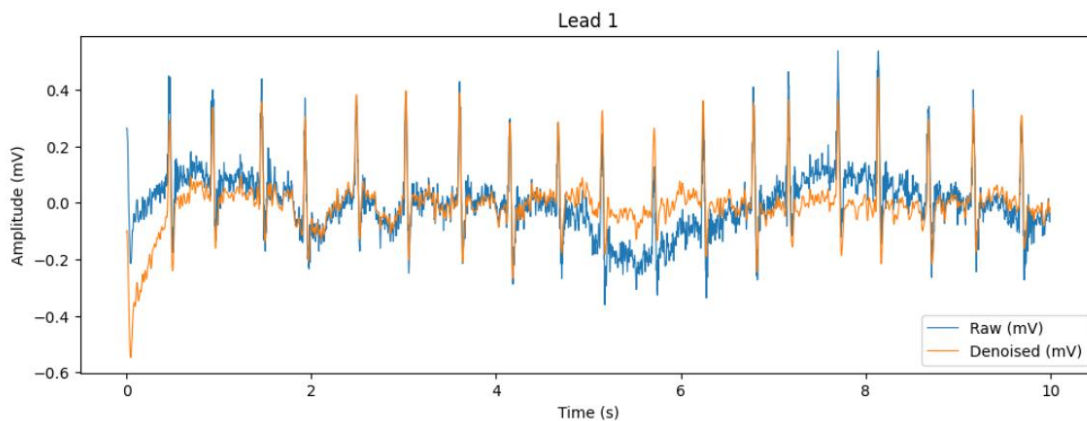
```
def apply_bandpass(sig, fs, lowcut=0.5, highcut=40.0, order=4):
    """Lọc giữ lại tần số sinh lý ECG (0.5–40Hz)"""
    b, a = butter_bandpass(lowcut, highcut, fs, order)
    return filtfilt(b, a, sig)
```

Chuyển đổi tín hiệu về mV:

```
def normalize_to_mV(sig, gain=1000):
    """Chuyển đổi từ đơn vị raw ADC sang mV"""
```

```
return sig / gain
```

Kết quả thu được trên một chuyển đạo mẫu với màu xanh là tín hiệu gốc, màu cam là tín hiệu sau xử lý:



Hình 20: Hình ảnh sóng thu được sau chuẩn hóa

Sau khi lọc, tín hiệu về gần đường đẳng điện hơn nằm gọn trên một đường thẳng không bị chênh lệch xuống quá nhiều như ban đầu mà vẫn giữ được hình dáng gốc của tín hiệu.

Để huấn luyện dữ liệu ta không thể để nguyên tín hiệu được mà ta buộc phải chuyển về dạng tín hiệu lưu đuôi .npy huấn luyện theo mô hình 1D-CNN hoặc trực quan hóa về dạng spectrogram huấn luyện theo mô hình 2D-CNN. Ở đây chúng em tiến hành theo hướng số một, tức là huấn luyện theo mô hình 1D-CNN, do đó, để huấn luyện thì sau bước lọc nhiễu tiến hành chuẩn hóa và lưu file về dưới dạng mảng số numpy:

- Chuẩn hóa 8 nhãn theo one-hot code đánh số từ [0-7]

```
import pandas as pd
import numpy as np
# Danh sách top 8 SNOMED CT phổ biến nhất
top_10_codes = ["426177001", "426783006", "164890007", "427084000",
                "427393009", "164889003", "429622005", "39732003"]
# Tạo cột multi-hot cho từng bệnh nhân
for code in top_10_codes:
    df_resolved[code] = df_resolved['diagnosis_codes'].apply(lambda x: 1 if
```

```

code in x.split(',') else 0) # sd bảng dữ liệu đã sửa
# Lấy X, y
X_paths = df_resolved['file_path'].values # danh sách file .mat
y = df_resolved[top_10_codes].values # shape: (num_samples, 10)
print(f'Shape y: {y.shape}') # (45152, 10)
print(f'Ví dụ nhãn 1 bệnh nhân: {y[0]}')

```

- Lọc nhiễu toàn bộ dữ liệu và stack lại thành mảng numpy:

```

import os
import numpy as np
signals_list = []
for path in X_paths:
    # đổi đuôi .hea -> .mat
    base, ext = os.path.splitext(path)
    mat_path = base + ".mat"
    # kiểm tra file .mat có tồn tại
    if not os.path.exists(mat_path):
        print(f'File .mat không tồn tại: {mat_path}')
        continue
    signals = load_ecg_mat(mat_path) # shape (12, N)
    fs = 500
    # Chuẩn hóa sang mV
    raw_mV = normalize_to_mV(signals, gain=1000)
    # Lọc nhiễu
    denoised = denoise_all_leads(raw_mV, fs)
    signals_list.append(denoised)
# Stack lại thành mảng numpy
X = np.stack(signals_list, axis=0)
print(f'X.shape = {X.shape}')

```

- Tạo thư mục mới để lưu dữ liệu:


```
save_dir = "processed_ecg_omaigah"  
os.makedirs(save_dir, exist_ok=True)
```

- Lưu lại file numpy vào thư mục

```
# Lưu lại file numpy  
    np.save(os.path.join(save_dir, f'record_{i}.numpy'), sig_z)  
    valid_indices.append(i)
```

Chương 4: Xây dựng và huấn luyện mô hình

4.1. Xây dựng mô hình ResNet 1D-CNN

Trước khi tiến hành xây dựng ta chia dữ liệu thành 3 tập train/val/test với train chiếm 70%, val 10% và test 20%. Tuy nhiên, số lượng file trong tập train, val và test hoàn toàn được chia ngẫu nhiên nên phải khớp với các nhãn bệnh mà trước đó ta đã one-hot ở phần tiền xử lý, nên sau khi lưu lại các file dưới dạng .npy ta sẽ mapping thêm lần nữa để các nhãn khớp với các tập ta đã chia, thuận tiện cho việc huấn luyện và dự đoán sau này:

- Chia dữ liệu sau khi lưu file:

```
def split_dataset(
    save_dir: str,
    df_resolved: pd.DataFrame,
    snomed_codes: list,
    test_size=0.3,
    val_ratio=2 / 3,
    random_state=42,
):
```

- Mapping labels với files trong tập đã chia:

```
y_train = np.array([y[get_index_from_filename(f)] for f in train_files])
y_val = np.array([y[get_index_from_filename(f)] for f in val_files])
y_test = np.array([y[get_index_from_filename(f)] for f in test_files])

print("y_train shape:", y_train.shape)
print("y_val shape:", y_val.shape)
print("y_test shape:", y_test.shape)
```

Trong quá trình xây dựng pipeline để huấn luyện, lường trước việc mất cân bằng giữa các nhãn dễ gây thiên lệch khiến mô hình tập trung nhận diện vào các nhãn có số dữ liệu lớn hơn nên ta tiến hành tạo thêm dữ liệu ngẫu nhiên để tăng độ đa dạng cho các nhãn có ít dữ liệu, như dịch trái/phải tín hiệu, thêm nhiễu nhẹ giữ lại hình dạng dữ liệu tới 90% so với ban đầu, bằng cách này dữ liệu vẫn giữ được tính chất của nó.

```
def augment_signal(self, sig):
    """Tạo các biến thể dữ liệu ngẫu nhiên để tăng độ đa dạng"""
    # Dịch trái/phải (temporal shift)
    shift = np.random.randint(-50, 50)
    sig = np.roll(sig, shift, axis=1)
    # Thêm nhiễu Gaussian nhẹ
    noise = np.random.normal(0, 0.01, sig.shape)
    sig += noise.astype(np.float32)
    # Scale biên độ
    scale = np.random.uniform(0.9, 1.1)
    sig *= scale
    # Clip lại để tránh overflow
    sig = np.clip(sig, -5, 5)
    return sig
```

Khi huấn luyện, ta chỉ cần bật `augment = True` để dữ liệu được tạo giả, tăng xác suất cho các lớp nhãn có dữ liệu ít trong tập train.

Ngoài ra, để mô hình nhận diện và học được đặc trưng sóng tốt hơn, nhất là với dữ liệu liên quan đến điện tâm đồ thì các đặc trưng của sóng rất quan trọng, việc xây dựng một mô hình để mô hình ấy tập trung vào những dạng sóng trong điện tâm đồ như sóng Q, T, QRS, U,... là cực kì cần thiết để chẩn đoán bệnh. Do đó ta sử dụng thêm Attention để học trọng số của từng “thời điểm” (time step) trong chuỗi đặc trưng đầu vào, tập trung hơn vào những đoạn tín hiệu chứa thông tin bệnh lý, thay vì coi mọi điểm thời gian đều quan trọng như nhau.

```

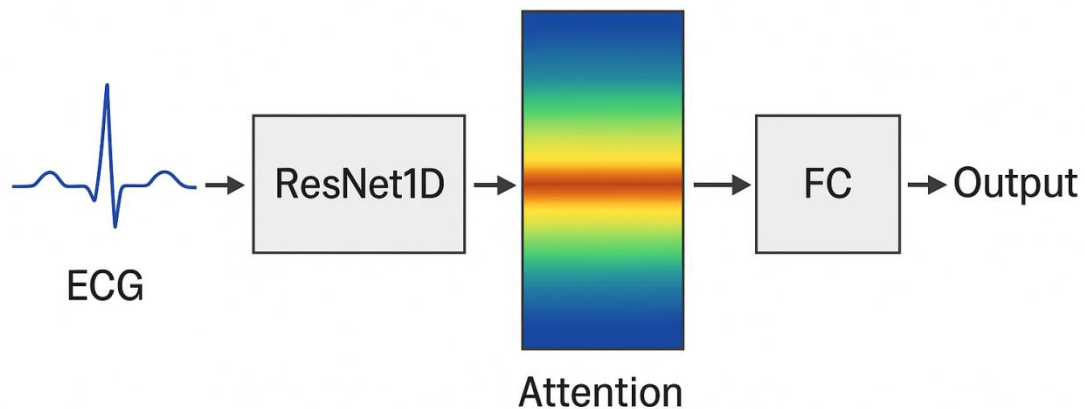
class Attention(nn.Module):
    def __init__(self, input_dim):
        super(Attention, self).__init__()
        self.attn = nn.Sequential(
            nn.Linear(input_dim, 128),
            nn.Tanh(),
            nn.Linear(128, 1)
        )
    def forward(self, x):
        # x: (batch, seq_len, feat_dim)
        weights = self.attn(x) # (batch, seq_len, 1)
        weights = F.softmax(weights, dim=1)
        context = torch.sum(weights * x, dim=1) # (batch, feat_dim)
        return context, weights

```

Trong đó:

- **weights = self.attn(x)**: tính điểm attention cho mỗi time step, kết quả trả về batch, seq_len (số điểm thời gian) => mỗi bước thời gian có 1 điểm trọng số
- **weights = F.softmax(weights, dim=1)**: chuẩn hóa thành xác suất với softmax [0,1]
- **context = torch.sum(weights * x, dim=1)**: lấy trung bình có trọng số, tổng hợp lại thành một vector duy nhất thể hiện nơi mô hình chú ý nhất.

Vậy tổng hợp lại, để huấn luyện được mô hình ta phải đi qua những bước sau, từ tín hiệu ecg lưu dưới dạng ecg -> đưa vào mô hình ResNet 1D-CNN -> dùng Attention để tập trung vào những điểm sóng quan trọng -> qua lớp Fully Connected Layer -> trả về giá trị dự đoán.



Hình 21: Xây dựng pipeline huấn luyện dữ liệu

Kiến trúc 1D-ResNet50:

```
# === Bottleneck Block cho ResNet50 1D ===
class Bottleneck1D(nn.Module):
    expansion = 4 # số nhân kênh (ResNet50 tiêu chuẩn)

    def __init__(self, in_channels, out_channels, stride=1, downsample=None):
        super(Bottleneck1D, self).__init__()
        self.conv1 = nn.Conv1d(in_channels, out_channels, kernel_size=1,
                                bias=False)
        self.bn1 = nn.BatchNorm1d(out_channels)
        self.conv2 = nn.Conv1d(out_channels, out_channels, kernel_size=3,
                                stride=stride, padding=1, bias=False)
        self.bn2 = nn.BatchNorm1d(out_channels)
        self.conv3 = nn.Conv1d(out_channels, out_channels * self.expansion,
                                kernel_size=1, bias=False)
        self.bn3 = nn.BatchNorm1d(out_channels * self.expansion)
        self.relu = nn.ReLU(inplace=True)
        self.downsample = downsample

    def forward(self, x):
        identity = x
```

```

if self.downsample is not None:
    identity = self.downsample(x)

    out = self.relu(self.bn1(self.conv1(x)))
    out = self.relu(self.bn2(self.conv2(out)))
    out = self.bn3(self.conv3(out))

    out += identity
    out = self.relu(out)
    return out

# === ResNet50 1D + Attention ===
class ResNet1D50Attention(nn.Module):
    def __init__(self, num_classes=8):
        super(ResNet1D50Attention, self).__init__()
        self.in_channels = 64

        # Khởi tạo conv đầu vào
        self.conv1 = nn.Conv1d(12, 64, kernel_size=7, stride=2, padding=3,
bias=False)
        self.bn1 = nn.BatchNorm1d(64)
        self.relu = nn.ReLU(inplace=True)
        self.maxpool = nn.MaxPool1d(kernel_size=3, stride=2, padding=1)

        # Các block ResNet50
        self.layer1 = self._make_layer(64, 3)
        self.layer2 = self._make_layer(128, 4, stride=2)
        self.layer3 = self._make_layer(256, 6, stride=2)
        self.layer4 = self._make_layer(512, 3, stride=2)

        # Attention layer

```

```

self.attention = Attention(512 * Bottleneck1D.expansion)

# Fully connected
self.fc = nn.Linear(512 * Bottleneck1D.expansion, num_classes)

def _make_layer(self, out_channels, blocks, stride=1):
    downsample = None
    if stride != 1 or self.in_channels != out_channels *
Bottleneck1D.expansion:
        downsample = nn.Sequential(
            nn.Conv1d(self.in_channels, out_channels *
Bottleneck1D.expansion,
                kernel_size=1, stride=stride, bias=False),
            nn.BatchNorm1d(out_channels * Bottleneck1D.expansion),
        )

    layers = [Bottleneck1D(self.in_channels, out_channels, stride,
downsample)]
    self.in_channels = out_channels * Bottleneck1D.expansion
    for _ in range(1, blocks):
        layers.append(Bottleneck1D(self.in_channels, out_channels))
    return nn.Sequential(*layers)

def forward(self, x):
    # Input: (batch, 12, seq_len)
    x = self.relu(self.bn1(self.conv1(x)))
    x = self.maxpool(x)

    x = self.layer1(x)
    x = self.layer2(x)
    x = self.layer3(x)

```

```

x = self.layer4(x)

# Attention
x = x.permute(0, 2, 1) # (batch, seq_len, feat_dim)
x, weights = self.attention(x)

out = self.fc(x)
return out

```

Với dữ liệu đầu vào là tín hiệu của 12 chuyển đạo đi qua các tầng tích chập, mỗi tầng sẽ giảm độ dài tín hiệu và tăng số lượng đặc trưng lên, giúp mô hình học sâu hơn qua từng tầng. Mô hình tạo dựa trên nguyên gốc ResNet50 nhận diện ảnh 2D do vậy kiến trúc của model rất sâu và phức tạp qua từng tầng. Ở tầng Fully Connected cuối cùng thì số in features sau khi flatten là 2048 đặc trưng, out features là 8 nhãn bệnh đã được lọc trước đó.

Để quá trình huấn luyện được chắc chắn hơn và quá trình mô hình học được đặc trưng từ những lớp có số dữ liệu ít, ta thêm bước tính trọng số cho mô hình, để mô hình tập trung mạnh hơn vào những dữ liệu này:

```

# --- Tính pos_weight theo công thức: (số mẫu âm) / (số mẫu dương) ---
num_samples, num_classes = y_train.shape
positive_counts = y_train.sum(axis=0)
negative_counts = num_samples - positive_counts
pos_weight = negative_counts / (positive_counts + 1e-6)
pos_weight = torch.tensor(pos_weight, dtype=torch.float32).to(device)
print("pos_weight:", pos_weight)

```

4.2. Xây dựng mô hình InceptionTime

Đối với mô hình Inception Time, các bước tăng cường dữ liệu, tính trọng số và chia dữ liệu tương tự như mô hình ResNet, ta chỉ thay đổi hàm xây dựng mô hình để huấn luyện, ta xây dựng mô hình này với cấu trúc độ sâu tương đồng với ResNet34:


```
import torch
import torch.nn as nn
from training.attention import Attention
```

4.3. Huấn luyện và đánh giá mô hình

Huấn luyện với mô hình ResNet50 ta được kết quả sau:

```
=== Đánh giá ResNet1D50 + Attention ===

Classification Report:
              precision    recall  f1-score   support

     0       0.97         0.98         0.98         3267
     1       0.83         0.94         0.88         1610
     2       0.72         0.99         0.83         1630
     3       0.94         0.97         0.95         1469
     4       0.19         0.88         0.31          482
     5       0.19         0.99         0.32          376
     6       0.21         0.87         0.34          336
     7       0.32         0.98         0.48          323

 micro avg       0.59         0.97         0.73         9493
 macro avg       0.55         0.95         0.64         9493
weighted avg       0.78         0.97         0.83         9493
 samples avg       0.69         0.95         0.77         9493
```

Hình 22: Bảng kết quả classification report ResNet50

Có thể thấy được những lớp có dữ liệu ít mang lại kết quả nhận diện không tốt, đổi lại tỉ lệ ở những lớp có lượng dữ liệu cao khá ổn và nhận diện tốt. Để cải thiện hơn, ta lưu model lại và tiếp tục tìm ngưỡng tốt nhất cho mô hình:

```
def find_best_threshold(y_true, y_pred_proba):
    thresholds = np.linspace(0.1, 0.9, 17) # các ngưỡng từ 0.1 đến 0.9, bước 0.05
    best_thresh = []
    for i in range(y_true.shape[1]): # chạy cho từng class
        best_t, best_f1 = 0.5, 0
        for t in thresholds:
            preds = (y_pred_proba[:, i] > t).astype(int)
```

```

f1 = f1_score(y_true[:, i], preds)
if f1 > best_f1:
    best_f1, best_t = f1, t
best_thresh.append(best_t)
return np.array(best_thresh)

```

Rồi dự đoán lại trên tập test, lúc này kết quả cho ra ổn định hơn ban đầu rất nhiều:

```

=== Hiệu suất sau khi tinh chỉnh threshold ===

```

	precision	recall	f1-score	support
0	0.98	0.98	0.98	3267
1	0.87	0.92	0.89	1610
2	0.77	0.96	0.86	1630
3	0.96	0.96	0.96	1469
4	0.27	0.60	0.37	482
5	0.23	0.82	0.36	376
6	0.37	0.66	0.47	336
7	0.52	0.72	0.61	323
micro avg	0.73	0.92	0.81	9493
macro avg	0.62	0.83	0.69	9493
weighted avg	0.82	0.92	0.86	9493
samples avg	0.79	0.92	0.83	9493

Hình 23: Bảng kết quả classification report ResNet50 sau khi chỉnh threshold

Có thể thấy micro avg tăng lên tới 0.8 tốt hơn so với lúc chưa tinh chỉnh ngưỡng, chỉ số precision ở những lớp ít cũng được cải thiện nhiều. Tuy nhiên kết quả này vẫn còn khá kém, vẫn cần phải tối ưu thêm.

Huấn luyện với mô hình InceptionTime ta được kết quả sau:

Classification Report:					
	precision	recall	f1-score	support	
0	0.89	0.98	0.93	3257	
1	0.77	0.87	0.82	1623	
2	0.68	0.97	0.80	1615	
3	0.85	0.95	0.90	1451	
4	0.20	0.66	0.31	492	
5	0.17	0.98	0.29	360	
6	0.18	0.86	0.30	335	
7	0.25	1.00	0.40	308	
micro avg	0.55	0.94	0.70	9441	
macro avg	0.50	0.91	0.59	9441	
weighted avg	0.72	0.94	0.79	9441	
samples avg	0.66	0.91	0.73	9441	

Hình 24: Bảng kết quả classification report InceptionTime

Ta cũng chỉnh lại threshold cho mô hình và được kết quả:

Classification Report:

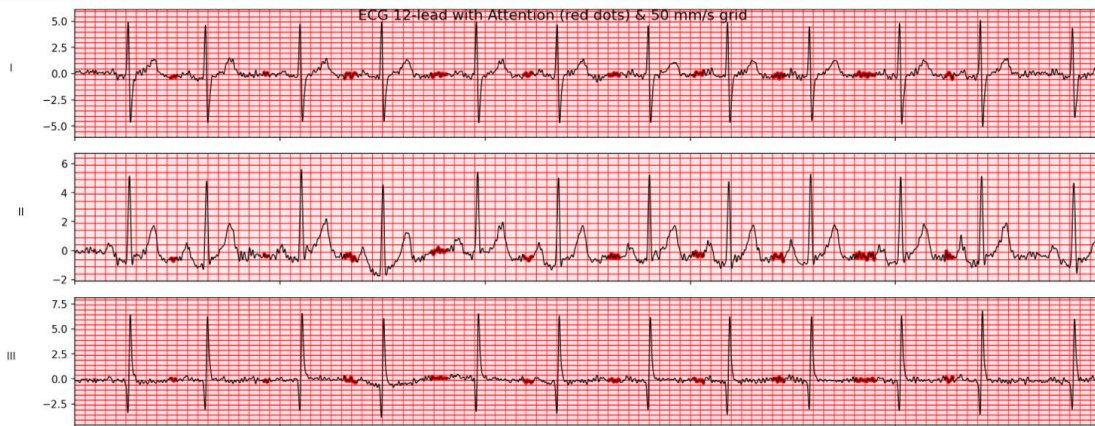
	precision	recall	f1-score	support
Nhịp chậm xoang	0.91	0.98	0.95	1632
Nhịp xoang bình thường	0.78	0.84	0.81	803
Cuồng nhĩ	0.71	0.92	0.80	813
Chênh lên đoạn ST	0.95	0.91	0.93	741
Loạn nhịp xoang	0.31	0.45	0.37	249
Rung nhĩ	0.20	0.91	0.32	176
Chênh xuống đoạn ST	0.41	0.55	0.47	182
Trục điện tim lệch trái	0.36	0.95	0.52	154
micro avg	0.68	0.89	0.77	4750
macro avg	0.58	0.82	0.65	4750
weighted avg	0.77	0.89	0.81	4750
samples avg	0.74	0.87	0.78	4750

Hình 25: Bảng kết quả classification report Inception Time sau khi chỉnh threshold

Có thể thấy micro avg cải thiện lên 0.7 tuy nhiên so sánh với ResNet50 thì InceptionTime kém hiệu quả hơn, cụ thể là ResNet50 micro avg ở F1 Score 0.81 trong khi InceptionTime chỉ có 0.77 sau khi tinh chỉnh threshold.

4.4. Đánh giá mô hình với Attention

Trong quá trình huấn luyện dữ liệu thì để xem dữ liệu huấn luyện có “nhìn đúng” vùng đặc trưng cần nhận diện không, nhất là với dữ liệu của ECG 12 chuyển đạo thì ta phải đặc biệt chú ý đến các vùng sóng để từ đó nhận diện được mô hình huấn luyện có tập trung vào vùng sóng bất thường không. Từ đó dựa vào kiến thức y khoa lâm sàng để đánh giá và chẩn đoán bệnh một cách chính xác. Do vậy lúc ta tính trọng số attention trong lúc huấn luyện, sau khi huấn luyện xong ta sẽ trực quan lại vùng đặc trưng được mô hình nhận diện ấy:



Hình 26: Minh họa trực quan lại đặc trưng với Attention

4.5. Kết quả mô hình

Từ số liệu thu được, tương ứng với những nhãn:

- 0: Sinus Bradycardia - Nhịp chậm xoang
- 1: Sinus Rhythm - Nhịp xoang bình thường
- 2: Atrial Flutter - Cuồng nhĩ
- 3: ST Elevation - Chênh lên đoạn ST
- 4: Sinus Arrhythmia - Loạn nhịp xoang
- 5: Atrial Fibrillation (alternate) - Rung nhĩ
- 6: ST drop down - Chênh xuống đoạn ST
- 7: Axis left shift - Trục điện tim lệch trái

Kết quả cho thấy bệnh nhân bị nhịp chậm xoang và những người có nhịp xoang bình thường có chỉ số dự đoán ổn định và cao nhất, tương ứng với số mẫu cũng chiếm tỉ lệ cao. Từ đây giảm dần, một phần do việc sắp xếp nhãn cũng dựa theo độ phổ biến của chúng nên số mẫu của từng nhãn cũng ít dần đi, theo đó mà độ nhận diện cũng dần kém hơn cho thiếu thốn về dữ liệu cũng như việc tinh chỉnh dù đã làm qua nhiều công đoạn vẫn khó có thể cải thiện lên được mức cao hơn.

Cũng thông qua quá trình trực quan lại dữ liệu với attention để nhìn vào vùng đặc trưng, có thể thấy những vùng đặc trưng được mô hình nhận diện cũng là những vùng sóng bất thường của những nhãn bệnh, tuy nhiên vẫn còn một số vùng sóng không cần thiết bị mô hình tập trung quá mức.

Chương 5: Xây dựng Web App nhận diện và dự đoán bệnh từ tín hiệu điện tâm đồ

5.1. Xây dựng FrontEnd và BackEnd

Hệ thống gồm 2 phần:

- **Backend** : Nơi xử lý mô hình và nhận diện dữ liệu được import vào từ frontend để cho ra kết quả dự đoán đa nhãn.
- **Frontend** : Nơi hiển thị dữ liệu và trình bày hóa trực quan bằng dự đoán nhãn cũng như hình ảnh 12 chuyển đạo trực quan dạng sóng.

Công nghệ sử dụng:

NodeJS – là một framework nhẹ cho phát triển ứng dụng web phía backend.

OpenCV – là một thư viện được thiết kế để giải quyết các vấn đề thị giác máy tính.

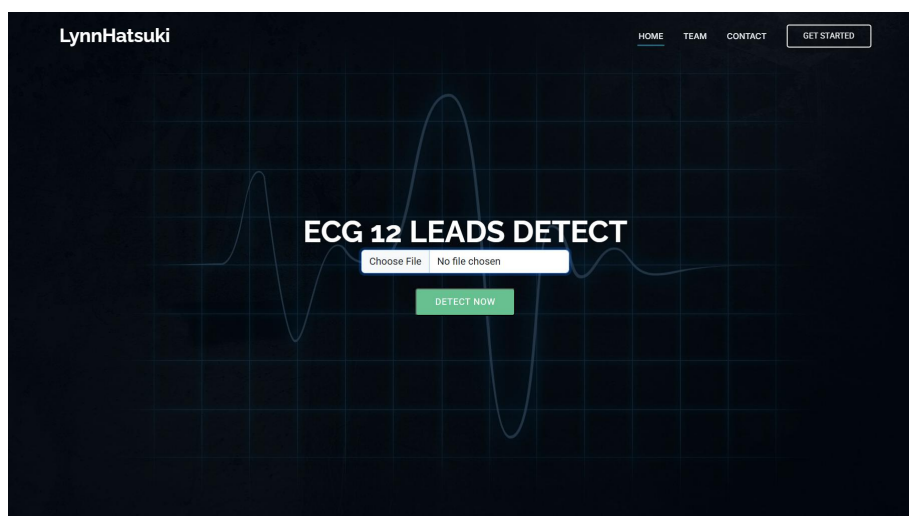
Pytorch – là một framework học máy, được tạo bởi Facebook.

NumPy – là thư viện xử lý số học và mảng n-chiều hiệu quả cho Python, cung cấp cấu trúc ndarray và hàng loạt phép toán đại số/tính toán ma trận.

SciPy (scipy.io) – module scipy.io dùng để đọc/ghi file định dạng khoa học, thuận tiện khi load dữ liệu y sinh/ECG từ file .mat.

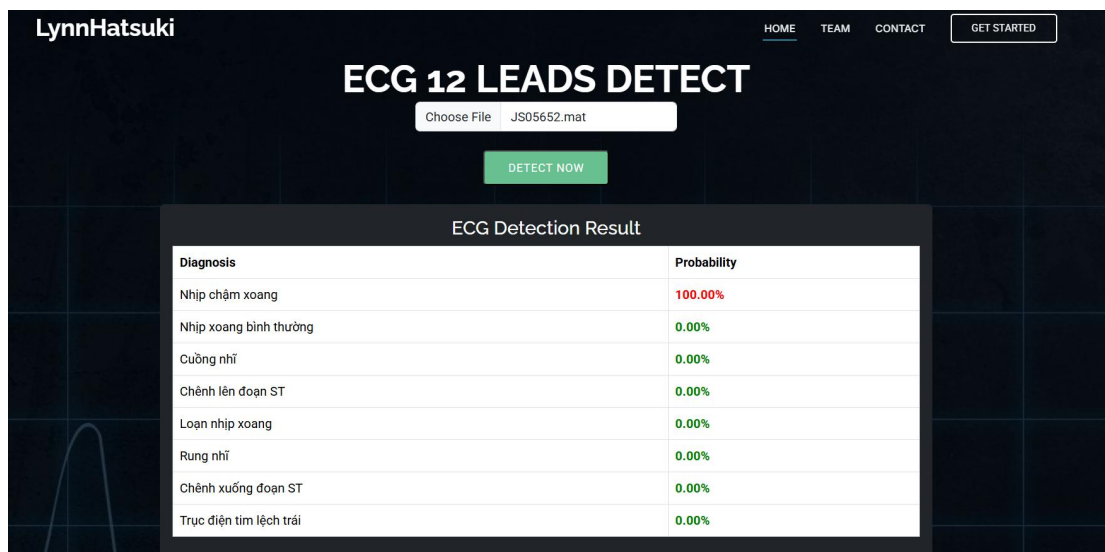
base64 – module mã hóa/giải mã Base64, thường dùng để chuyển hình ảnh/nhị phân thành chuỗi ASCII (để nhúng ảnh vào JSON giúp hiển thị ảnh 12 chuyển đạo qua frontend).

Hình ảnh giao diện trang web:



Hình 27: Giao diện Web App

Import .mat chọn Detect để hiển thị bảng kết quả nhận diện:



Hình 28: Giao diện dự đoán kết quả đa nhãn

Tích hợp xem ảnh 12 đạo trình:



Hình 29: Giao diện hiển thị kết quả ảnh 12 chuyển đạo

5.2. Cấu hình để load mô hình nhận diện sóng điện tâm đồ

Để có thể sử dụng mô hình trên Web App, ta cần làm theo những bước sau:

- Định nghĩa mô hình ResNet 1D-CNN
- Xử lý chính và dự đoán: ta định nghĩa mô hình và tải lên các trọng số bias, của mô hình.
- Định nghĩa hàm xử lý tín hiệu đầu vào, lọc nhiễu cho sóng giống như hàm đã định nghĩa khi huấn luyện trên notebook.

- Cấu hình thêm hàm hiển thị 12 đạo trình trên giao diện frontend với base64:

```
def plot_ecg_12leads(signals, fs=500):
    """Vẽ 12 đạo trình ECG riêng lẻ và trả về danh sách ảnh base64"""
    leads = [
        "I", "II", "III", "aVR", "aVL", "aVF",
        "V1", "V2", "V3", "V4", "V5", "V6"
    ]
    images = []
    t = np.arange(signals.shape[1]) / fs
    for i in range(12):
        fig, ax = plt.subplots(figsize=(6, 2))
        ax.plot(t, signals[i, :], linewidth=0.8)
        ax.set_title(leads[i])
        ax.set_ylabel("mV")
        ax.set_xlabel("Time (s)")
        ax.grid(True, linestyle="--", linewidth=0.5)
        buf = io.BytesIO()
        plt.tight_layout()
        plt.savefig(buf, format="png", dpi=120)
        plt.close(fig)
        buf.seek(0)
        img_b64 = base64.b64encode(buf.read()).decode("utf-8")
        images.append(img_b64)
    return images
```


III. TỔNG KẾT

1. Kết luận chung về đề tài

Đề tài đã xây dựng thành công mô hình học sâu (Deep Learning) nhằm phân loại tín hiệu điện tâm đồ (ECG) thành nhiều nhịp tim khác nhau, trong đó tập trung vào các nhịp bất thường như rung nhĩ (Atrial Fibrillation), nhịp nhanh, nhịp chậm, block nhánh,...

Các nội dung chính đã thực hiện gồm:

- Tiền xử lý dữ liệu ECG, bao gồm chuẩn hóa, cắt/đệm tín hiệu, xử lý nhiễu và cân bằng lớp.
- Xây dựng các mô hình CNN1D và ResNet1D kết hợp Attention, giúp tăng khả năng trích xuất đặc trưng theo cả không gian và thời gian.
- Áp dụng kỹ thuật data augmentation và pos_weight trong hàm loss để cải thiện hiệu suất với các lớp hiếm.
- Tiến hành cross-validation 5 lần, chọn mô hình tốt nhất, sau đó tối ưu ngưỡng (threshold) cho từng lớp để cân bằng giữa Precision và Recall.
- Sử dụng Grad-CAM 1D để giải thích mô hình, giúp quan sát vùng tín hiệu mà mô hình chú ý khi đưa ra quyết định chẩn đoán.

Kết quả cho thấy mô hình đạt được hiệu suất ổn định và khả năng nhận diện tốt đối với phần lớn nhịp tim, đặc biệt là các lớp phổ biến. Đối với các lớp hiếm, việc điều chỉnh threshold và trọng số lớp đã giúp cải thiện đáng kể chỉ số F1-score.

Grad-CAM giúp minh họa được cơ chế "chú ý" của mô hình, khẳng định mô hình thực sự học được các đặc trưng có ý nghĩa lâm sàng (như sóng P, QRS, T,...).

2. Hướng phát triển của đề tài

Trong tương lai, đề tài có thể được mở rộng theo các hướng sau:

- Mở rộng dữ liệu huấn luyện:
 - Bổ sung thêm nhiều nguồn ECG khác nhau, đa dạng bệnh nhân và tần số lấy mẫu để tăng tính tổng quát.
 - Thực hiện chuẩn hóa liên thiết bị (cross-device normalization) cho dữ liệu từ nhiều máy ECG khác nhau.

- Nâng cao mô hình:
 - Áp dụng các mô hình tiên tiến hơn như Transformer 1D, ECG-BERT hoặc Temporal Convolutional Networks (TCN).
 - Thử nghiệm mô hình lai CNN + LSTM/GRU để tận dụng cả khả năng trích đặc trưng cục bộ và phụ thuộc dài hạn.
- Phân tích giải thích sâu hơn:
 - Kết hợp Grad-CAM với Layer-wise Relevance Propagation (LRP) hoặc SHAP để hiểu rõ hơn quyết định của mô hình.
 - Đánh giá vùng “chú ý” trên từng lead ECG để đối chiếu với chẩn đoán của bác sĩ tim mạch.
- Ứng dụng thực tế:

Triển khai mô hình trong ứng dụng hỗ trợ chẩn đoán tim mạch hoặc thiết bị áo đo 12 chuyển đạo tim.

TÀI LIỆU THAM KHẢO

- [1] Ao, R., & He, G. (2023). Image based deep learning in 12-lead ECG diagnosis. *Frontiers in Artificial Intelligence*, 5.
<https://doi.org/10.3389/frai.2022.1087370>
- [2] Chang, P.-C., Liu, Z.-Y., Huang, Y.-C., Chen, J.-S., Chou, C.-C., Wo, H.-T., Lee, W.-C., Liu, H.-T., Wang, C.-C., Lin, C.-H., Tung, P.-H., Kuo, C.-F., & Wen, M.-S. (2025). Utilizing 12-lead electrocardiogram and machine learning to retrospectively estimate and prospectively predict atrial fibrillation and stroke risk. *Computers in Biology and Medicine*, 188, 109871. <https://doi.org/10.1016/j.compbiomed.2025.109871>
- [2] *ECG Learning Center - An introduction to clinical electrocardiography.* (n.d.). Ecg.utah.edu. <https://ecg.utah.edu/lesson/1>
- [3] Jackson, M. (2011, February 28). *How to Read an ECG*. Geeky Medics; Geeky Medics. <https://geekymedics.com/how-to-read-an-ecg/>
- [4] K, U. (2020). *Real-Time ECG QRS Detection - MATLAB & Simulink*. Wwww.mathworks.com. <https://www.mathworks.com/help/dsp/ug/real-time-ecg-qrs-detection.html>
- [5] K, U. (2021). *Electrocardiography - Cardiovascular Disorders*. MSD Manual Professional Edition.
<https://www.msdmanuals.com/professional/cardiovascular-disorders/cardiovascular-tests-and-procedures/electrocardiography>
- [6] Mec, V. (2024). *Hình dạng và cơ chế hình thành các sóng trên điện tâm đồ*. Vinmec International Hospital. <https://www.vinmec.com/vie/bai-viet/hinh-dang-va-co-che-hinh-thanh-cac-song-tren-dien-tam-do-vi>

- [7] Nakayama, M., Yagi, R., & Goto, S. (2025). Deep Learning Applications in 12-lead Electrocardiogram and Echocardiogram. *JMA Journal*, 8(1), 102–112. <https://doi.org/10.31662/jmaj.2024-0195>
- [8] Yousra Chahinez Hadj Azzem, & Fouzi Harrag. (2023). *Explainable Deep Learning Based-System for Multilabel Classification of 12-Lead ECG*. 1–6. <https://doi.org/10.1109/icnas59892.2023.10330482>