

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA TOÁN - TIN HỌC



Nhận dạng mẫu
Báo cáo đồ án giữa kì

Thành viên

| | |
|-----------------------|----------|
| Lê Quốc An | 22280001 |
| Nguyễn Công Tiến Dũng | 22280014 |
| Nguyễn Thiên Phúc | 22280067 |
| Lê Hoàng Nguyên | 22280061 |

Ngày 04, tháng 12, năm 2024

| No | Name | Id | Contribution |
|----|-----------------------|----------|--------------|
| 1 | Lê Quốc An | 22280001 | 100% |
| 2 | Nguyễn Công Tiên Dũng | 22280014 | 100% |
| 3 | Nguyễn Thiên Phúc | 22280067 | 100% |
| 4 | Lê Hoàng Nguyên | 22280061 | 100% |

Mục lục

| | |
|--|----|
| Mục lục..... | 2 |
| 1 Khái quát bộ dữ liệu..... | 3 |
| 2 Vấn đề và thử thách xung quanh bộ dữ liệu EEG..... | 4 |
| 3 Sự phân bố của dữ liệu theo từng kênh..... | 5 |
| 4 Data processing..... | 7 |
| 4.1 Trích xuất dữ liệu..... | 7 |
| 4.2 Kiểm tra dữ liệu không đủ điều kiện và loại bỏ chúng..... | 7 |
| 4.3 Áp dụng bộ lọc Band-pass và Notch..... | 7 |
| 4.4 Áp dụng ICA vào mô hình..... | 9 |
| 5 Feature Engineering..... | 11 |
| 5.1 Tính toán năng lượng phô cho dữ liệu..... | 11 |
| 5.2 Gộp bins tần số..... | 13 |
| 5.3 Tính trung bình qua cửa sổ chạy..... | 13 |
| 5.4 Chuẩn hoá biến đổi Logarit..... | 13 |
| 6 Label and Split Data..... | 14 |
| 6.1 Khởi tạo nhãn..... | 14 |
| 6.2 Chia dữ liệu..... | 14 |
| 7 Model theory..... | 14 |
| 7.1 SVM..... | 14 |
| 7.2 Random Forest..... | 15 |
| 8 Model Built..... | 16 |
| 8.1 SVM..... | 16 |
| 8.2 Random Forest..... | 17 |
| 9 Performance Evaluation..... | 17 |
| 9.1 ROC curves for model evaluation..... | 17 |
| 9.2 Model Selection..... | 18 |
| 9.3 Confusion matrices..... | 18 |
| 10 Hướng giải quyết cho vấn đề dự đoán của các model ra quá cao..... | 18 |
| 11 CNN (Đè xuát mở rộng)..... | 22 |

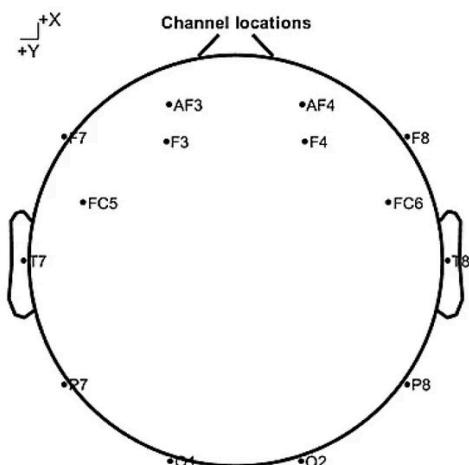
1 Khái quát bộ dữ liệu

Bộ dữ liệu EEG gồm 34 file .mat được thu thập từ một thí nghiệm điều khiển tàu mô phỏng bằng phần mềm “Microsoft Train Simulator”. Dữ liệu được lấy từ 5 người tham gia với tần số mẫu là 128Hz, trong đó:

- 5 người thực hiện, 7 lần đo mỗi người (2 lần đo đầu là thử nghiệm).
- Mỗi file là 1 lần đo của 1 người.

Mỗi file chứa 25 cột dữ liệu, với các cột tín hiệu chính từ kênh 4 đến kênh 17 tương ứng các kênh :

ED_AF3, ED_F7, ED_F3, ED_FC5, ED_T7, ED_P7, ED_O1, ED_O2, ED_P8, ED_T8,
ED_FC6, ED_F4, ED_F8, ED_AF4.



Thời gian mỗi thí nghiệm:

- Kéo dài từ 35-55 phút.

Chia làm 3 giai đoạn:

- **10 phút đầu:** Người tham gia tập trung giám sát đoàn tàu.
- **10 phút giữa:** Người tham gia ngừng chú ý nhưng vẫn tỉnh táo.
- **10 phút cuối:** Người tham gia thư giãn và có thể ngủ gật.

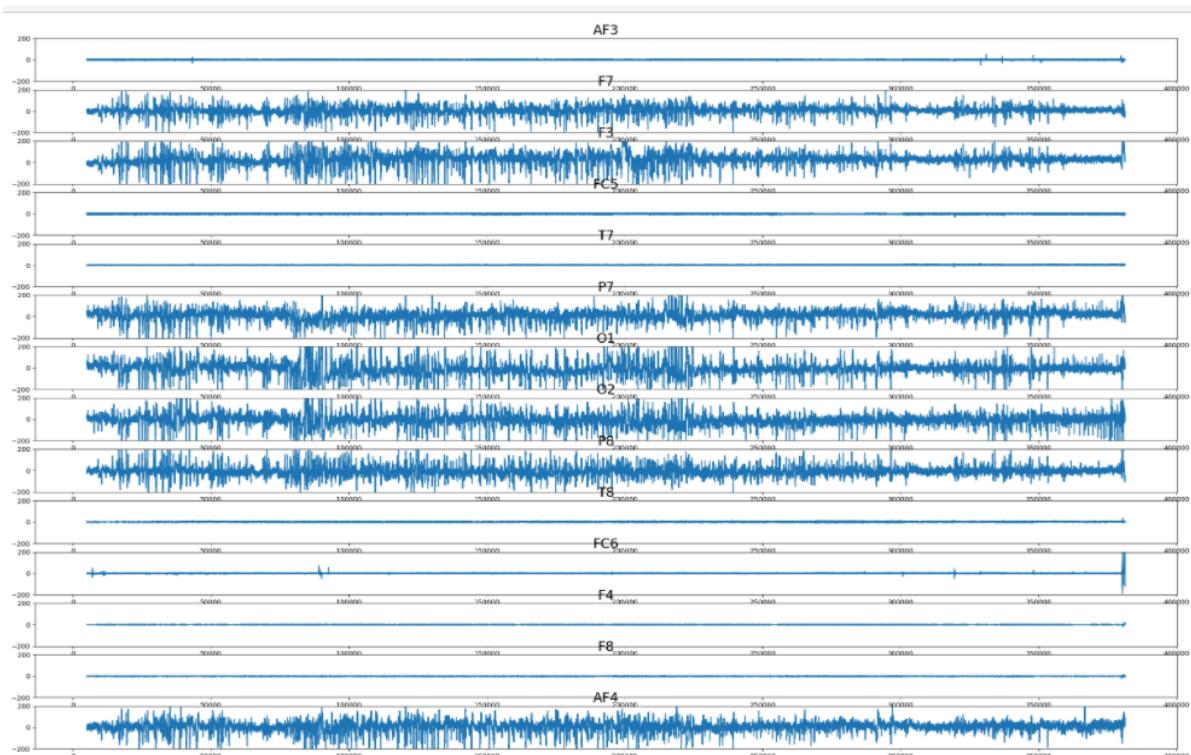
2 Vấn đề và thử thách xung quanh bộ dữ liệu EEG

| Dữ liệu EEG | Thử thách | Vấn đề | Giải pháp |
|--|--|---|---|
| Chất lượng và độ nhiễu của dữ liệu EEG | Dữ liệu EEG thường bị ảnh hưởng bởi nhiều từ môi trường bên ngoài và chuyển động của người tham gia. | Các kênh tín hiệu (từ 4 đến 17) có thể chứa nhiều từ tín hiệu cơ (EMG), chuyển động đầu hoặc nhiễu thiết bị. | Cần áp dụng các kỹ thuật lọc tín hiệu như lọc băng thông (bandpass filter) để loại bỏ nhiễu ngoài dải 0.5-40Hz và Notch filter (Bộ lọc hép) tại 50Hz để tránh nhiễu từ mạng lưới điện hoặc các thiết bị điện. |
| Phân biệt trạng thái tinh thần | Ba trạng thái (tập trung, mắt tập trung và buồn ngủ) có sự khác biệt nhỏ về mặt tín hiệu, đặc biệt ở trạng thái “mắt tập trung” khi không có dấu hiệu rõ ràng. | Khó khăn trong việc xác định trạng thái mắt tập trung vì không có biểu hiện vật lý như nhắm mắt hay thay đổi cử động. | Sử dụng các đặc trưng từ miền tần số (sóng alpha, beta,...) và các thuật toán học máy như SVM, Random Forest, mạng nơ-ron,... để phân loại chính xác. |
| Đồng bộ và thời gian thực | Dữ liệu phải được đồng bộ với thời gian thực từ các cột thời gian | Sai lệch thời gian hoặc lỗi đồng bộ có thể ảnh hưởng đến việc phân tích trạng thái tinh thần theo các giai đoạn. | Kiểm tra và chuẩn hóa dữ liệu thời gian trước khi xử lý, đảm bảo tính nhất quán giữa các lần đo |
| Dữ liệu không đồng đều giữa người tham gia | Một người chỉ có 1 lần đo, trong khi những người khác có 7 lần đo (2 lần thử nghiệm) | Sự không đồng đều về dữ liệu có thể gây khó khăn cho việc huấn luyện mô hình, đặc biệt là khi cần dữ liệu đồng nhất. | Cân nhắc loại bỏ dữ liệu không đầy đủ hoặc sử dụng kỹ thuật tăng cường dữ liệu (data augmentation). |

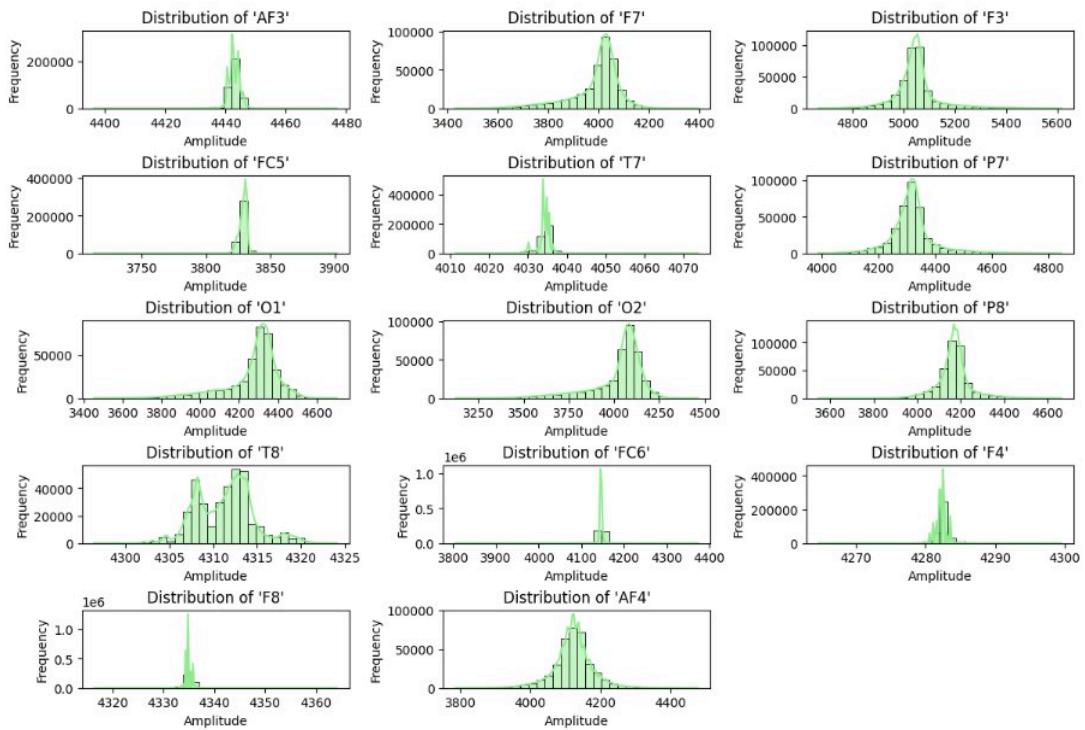
| | | | |
|--------------------------|--|--|--|
| Thời gian thí nghiệm dài | Mỗi thí nghiệm kéo dài từ 35 đến 55 phút, tạo ra lượng dữ liệu lớn và có thể gây khó khăn trong việc xử lý và lưu trữ. | Cần tối ưu hóa việc phân tích và lưu trữ dữ liệu khỏi lượng lớn. | Sử dụng các phương pháp nén dữ liệu hoặc xử lý dữ liệu theo lô (batch processing). |
|--------------------------|--|--|--|

3 Sự phân bố của dữ liệu theo từng kênh

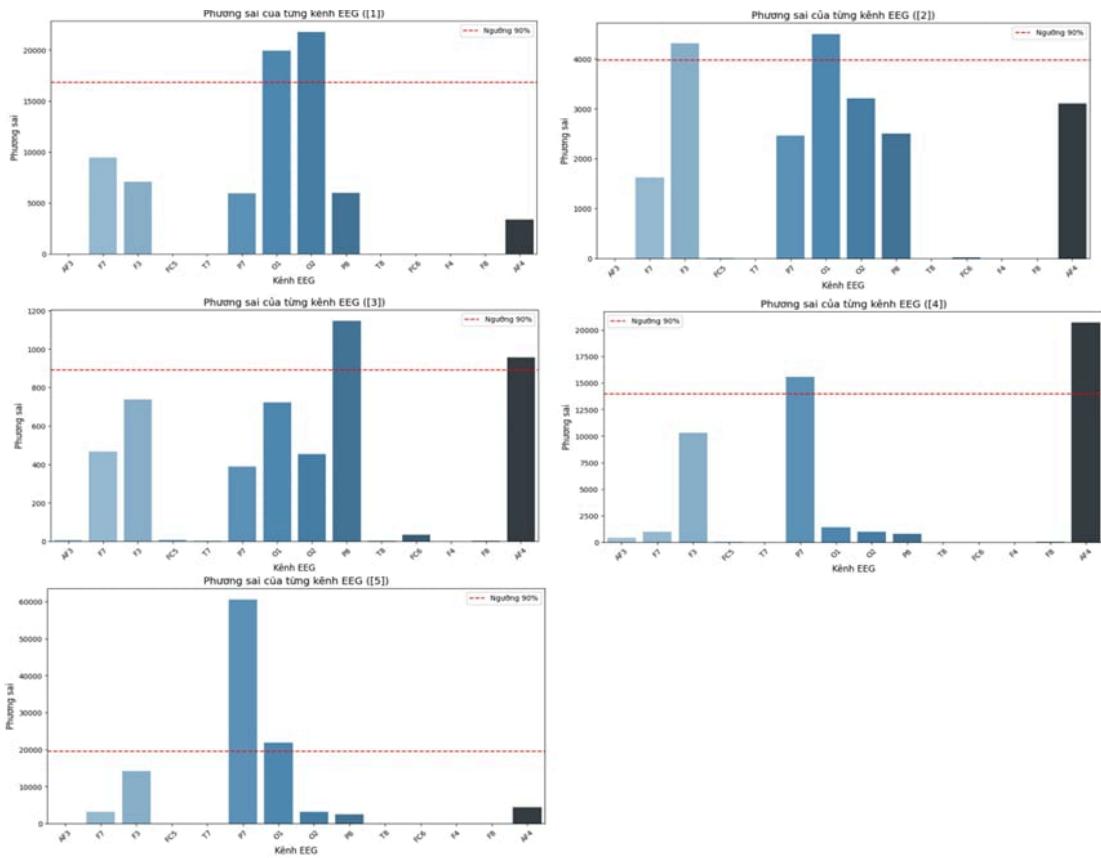
Nhận xét: Ta nhận thấy dữ liệu của 7 kênh: F7, F3, P7, O1, O2, P8, AF4 có variance rộng và có sự phân tán nên biên độ giao động sẽ có ảnh hưởng nhiều hơn so với những kênh còn lại. Mà theo như bài báo của tác giả có đề cập: "The acquired data from the 7 leads were identified by F3, F4, Fz, C3, C4, Cz, and Pz.". Hay 7 kênh mà tác giả muốn đề cập là F7, F3, P7, O1, O2, P8, AF4. Vậy nên ta sẽ chọn 7 kênh này và bỏ dữ liệu từ 7 kênh còn lại.



Hình 1. Độ biến thiên biên độ của từng kênh



Hình 2. Sự phân bố dữ liệu của từng kênh



Hình 3. Phương sai của từng người theo kênh trong 1 ngày

Ta có thể thấy từ nhiều ảnh trên có nhiều cột có phương sai cực kì nhỏ trong cả 5 tấm nên rất dễ gây khả năng phân biệt kém cho mô hình, dễ bị nhiễu.

4 Data processing

4.1 Trích xuất dữ liệu

Ý tưởng: Nhóm em sẽ tạo 5 Subject_files riêng cho 5 người và gắn lần lượt các file matlab theo từng ngày, riêng người cuối cùng chỉ có 4 ngày.

Đọc dữ liệu từ file và chỉ chọn 7 kênh F7, F3, P7, O1, O2, P8, AF4 để tạo các Subject_files, dưới đây là ví dụ của Subject_files tương ứng dữ liệu của người tham gia đầu tiên và các ngày thử nghiệm:

```
Subject_1:  
    Day_1: (357224, 7)  
    Day_2: (380344, 7)  
    Day_3: (351204, 7)  
    Day_4: (288752, 7)  
    Day_5: (398816, 7)
```

Giải thích: người Subject_1 ngày 1 có (357224,7) nghĩa là có 357224 mẫu trên 7 kênh.

4.2 Kiểm tra dữ liệu không đủ điều kiện và loại bỏ chúng

Công thức:

$$\text{Thời gian (phút)} = \frac{\text{Số mẫu}}{f_s \cdot 60}$$

Trong đó: $f_s = 128$ hz là tần số lấy mẫu

Kiểm tra xem dữ liệu trong từng ngày kiểm tra có đủ tối thiểu 30 phút hay không:

```
{'Subject_4': [( 'Day_5', 27.934895833333332)]}
```

4.3 Áp dụng bộ lọc Band-pass và Notch

Bộ lọc Band-pass: Bộ lọc dải thông cho phép tín hiệu trong một dải tần công cụ có thể đi qua và chặn tín hiệu ở các tần số ngoài dải đó. Ví dụ: một bộ lọc phạm vi thông tin có thể được sử dụng để giữ tín hiệu trong khoảng từ 1 Hz đến 50Hz, đồng thời loại bỏ tín hiệu ở tần số thấp

hơn 1 Hz và cao hơn 50 Hz. Bộ lọc này thường được sử dụng để loại bỏ các tần số không mong muốn nhưng vẫn giữ lại các tần số quan trọng.

Bộ lọc Notch: Bộ lọc hẹp được thiết kế để loại bỏ tín hiệu ở một tần số duy nhất hoặc một dải tần số rất hẹp. Một ví dụ phổ biến là loại bỏ tần số 50Hz (hoặc 60Hz tùy vào quốc gia) từ tín hiệu điện do nguồn điện lưới gây ra (tần số nhiễu). Bộ lọc Notch "cắt" (hoặc làm suy giảm mạnh) tín hiệu ở tần số cụ thể mà không ảnh hưởng nhiều đến các tần số xung quanh.

Lợi ích khi áp dụng vào model:

Bộ lọc Notch tại 50Hz: Dữ liệu EEG thường bị nhiễu từ các nguồn điện lưới (tần số 50Hz ở nhiều quốc gia). Việc áp dụng bộ lọc notch giúp loại bỏ nhiễu này mà không làm ảnh hưởng đến các tần số quan trọng trong dải tần của EEG.

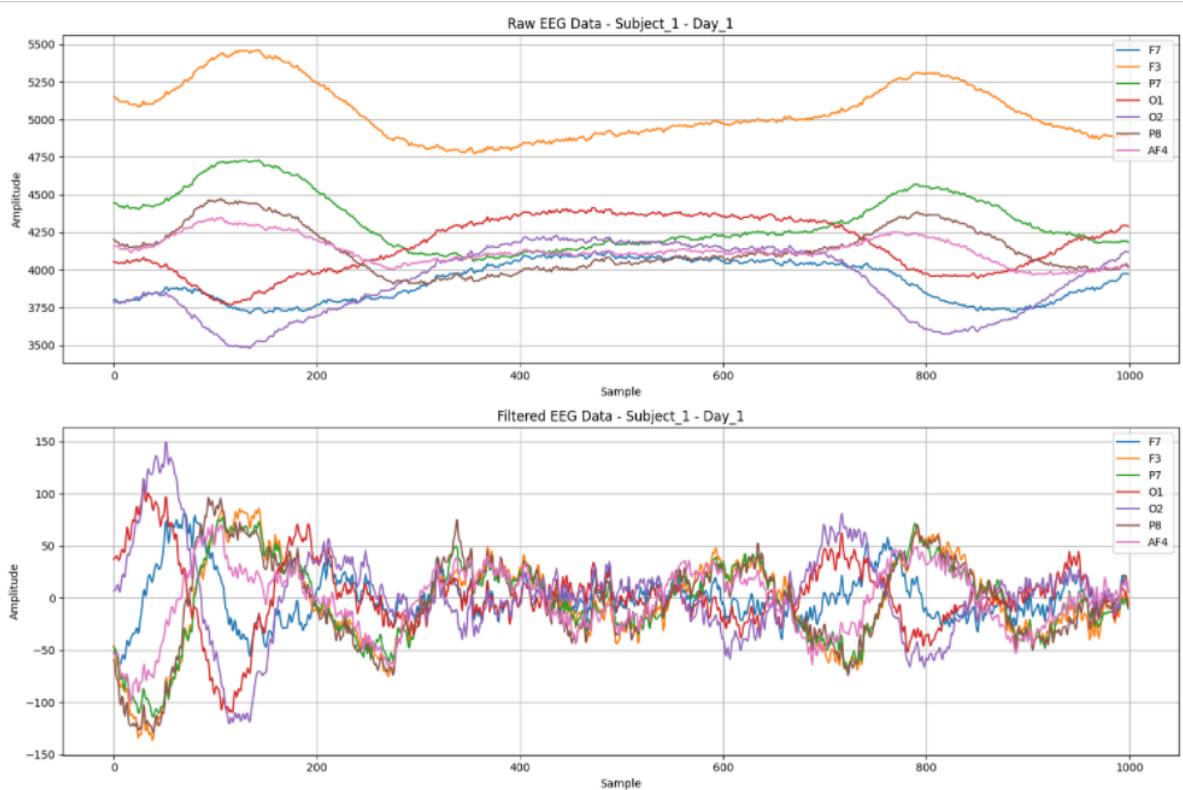
Bộ lọc Band-pass (0.5Hz đến 40Hz): Bộ lọc này giúp loại bỏ các tần số rất thấp (có thể là nhiễu hoặc tín hiệu nền) và tần số cao (có thể là nhiễu từ các thiết bị khác hoặc tín hiệu không liên quan). Điều này giúp bảo tồn các tần số quan trọng của sóng não, ví dụ như sóng delta, theta, alpha, beta, và gamma, mà mô hình EEG cần để phân tích.

Lý do chọn:

Đo tín hiệu EEG chứa sóng não phô biến trong đoạn 0.5 tới 40. Ví dụ: Delta (0.5 - 4 Hz), Theta (4 - 8 Hz), Alpha (8 - 13 Hz), Beta (13 - 30 Hz), Gamma (30 - 40 Hz)

Và thường trong môi trường thực tế, tín hiệu EEG thường bị nhiễu từ hệ thống điện lưới, đặc biệt là tại tần số 50 Hz (ở nhiều quốc gia) hoặc 60Hz (ở các quốc gia khác). Nี่ xuất phát từ các thiết bị điện tử xung quanh, chẳng hạn như máy tính, đèn chiếu sáng, hoặc các thiết bị điện khác.

So sánh kết quả giữa trước khi lọc và sau khi lọc:



Hình 4. Người thứ 1 vào ngày đầu tiên

Nhận xét: Ta thấy rằng trước khi lọc tín hiệu khá là rời rạc với nhau sau khi lọc tín hiệu

Dữ liệu thô: Các sóng EEG dao động mạnh và có nhiều nhiễu, khiến tín hiệu không đồng nhất.

Dữ liệu đã lọc: Sau khi lọc, các sóng EEG được gộp lại mượt mà hơn, giúp loại bỏ các nhiễu và tăng cường tín hiệu cần thiết. Tuy nhiên, một số sóng có thể bị gộp lại do quá trình lọc, điều này có thể làm mất đi một số chi tiết nhỏ nhưng giúp tín hiệu chính trở nên rõ ràng hơn.

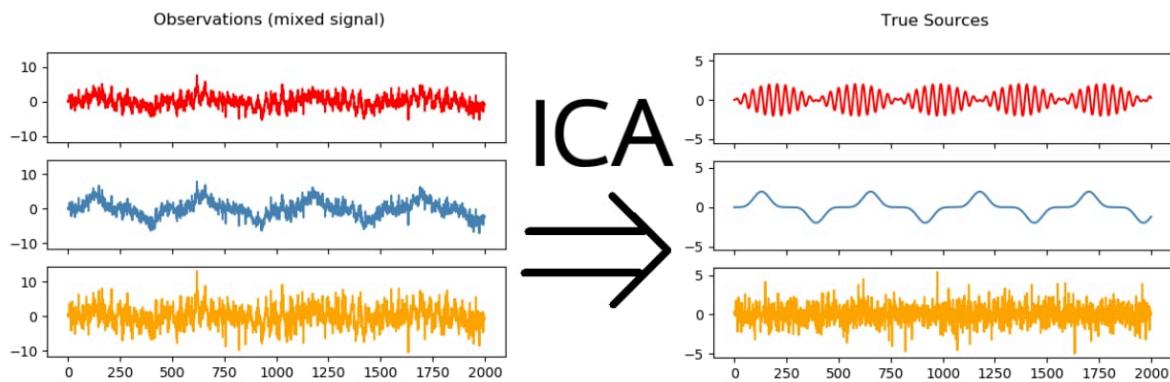
4.4 Áp dụng ICA vào mô hình

ICA (Independent Component Analysis) là một phương pháp phân tích tín hiệu, được sử dụng để tách các thành phần độc lập trong một tập hợp tín hiệu hỗn hợp.. Dưới đây là một số lợi ích chính của ICA khi áp dụng vào mô hình này:

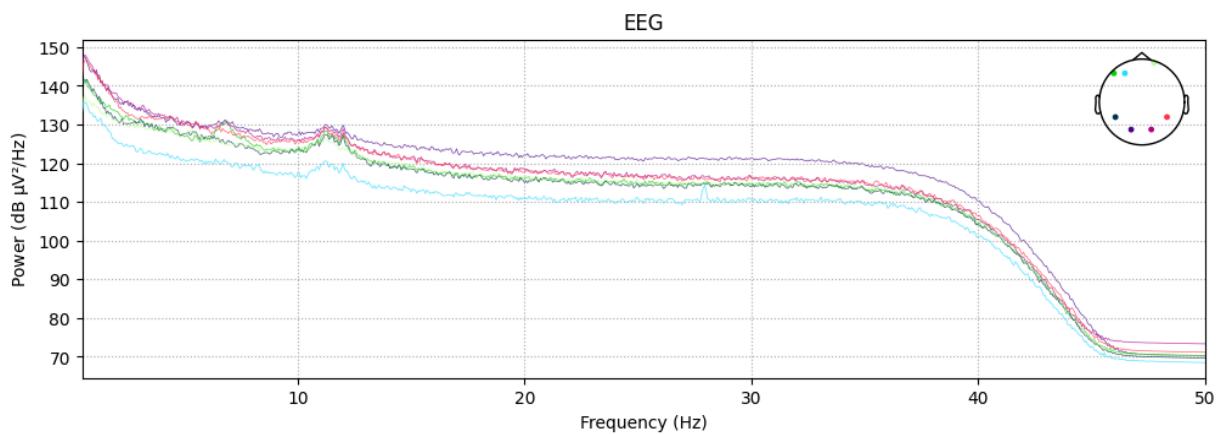
Loại bỏ nhiễu: Nhiều từ mắt (EOG), Nhiều cơ (EMG), Nhiều điện tử

Tối ưu hóa mô hình học máy: Dữ liệu sạch và không có nhiễu giúp tăng cường độ chính xác của các mô hình học máy như CNN, SVM, hay các mô hình phân loại khác

Giảm độ phức tạp của dữ liệu: ICA giúp giảm số lượng thành phần không cần thiết và chỉ giữ lại những thông tin quan trọng, giúp giảm độ phức tạp của mô hình và cải thiện hiệu suất tính toán.



Hình 5. Áp dụng ICA để tách tín hiệu



Hình 6. Biểu đồ mật độ phổ công suất (PSD)

Nhận xét: các đường có xu hướng giảm dần khi tần số tăng, cho thấy năng lượng của tín hiệu tập trung nhiều hơn ở các tần số thấp và giảm mạnh ở tần số cao.

Năng lượng cao nhất nằm trong khoảng từ 0 đến 10 Hz, tương ứng với các dải sóng delta (0.5-4 Hz) và theta (4-8 Hz).

Từ 10Hz đến khoảng 40 Hz, năng lượng giảm dần và có xu hướng ổn định.

Sau 40 Hz, năng lượng giảm mạnh, cho thấy đã có sự lọc bỏ các tần số cao hơn 40 Hz.

Một đường thẳng đứng có thể ám chỉ sự cắt tần số tại 40Hz, thể hiện việc áp dụng bộ lọc band-pass.

Phân chia dữ liệu:

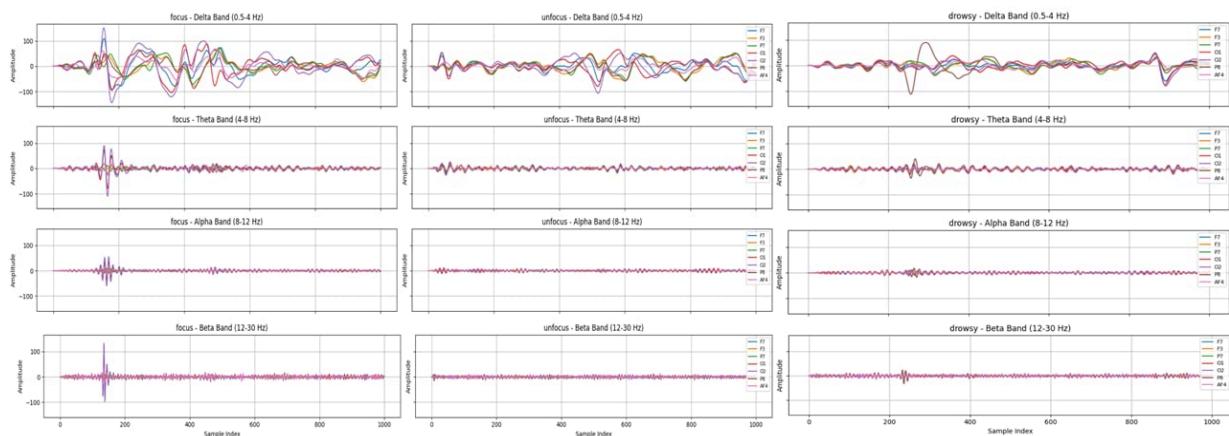
Chia 5 người thành 5 danh sách dữ liệu khác nhau và trong mỗi người sẽ chia 10 phút đầu là tập trung, 10 phút sau không tập trung, 10 phút kế là buồn ngủ của từng file được sắp xếp sẵn

So sánh trên các dải tần số khác nhau

Chúng em chọn 4 dải tần số:

'Delta': (0.5, 4), 'Theta': (4, 8), 'Alpha': (8, 12), 'Beta': (12, 30)

Kiểm tra so sánh trên 3 trạng thái (focus, unfocus, drowsy)



Hình 7. So sánh biên độ dao động tần số của Focus(bên trái), Unfocus(ở giữa) và Drowsy(bên phải)

5 Feature Engineering

5.1 Tính toán năng lượng phổ cho dữ liệu

Mục đích: Áp dụng STFT để phân tích tín hiệu EEG, giúp chuyển đổi tín hiệu thời gian (time-domain) thành tín hiệu tần số (frequency-domain), từ đó có thể tính toán năng lượng phổ (spectral power).

Cách thức:

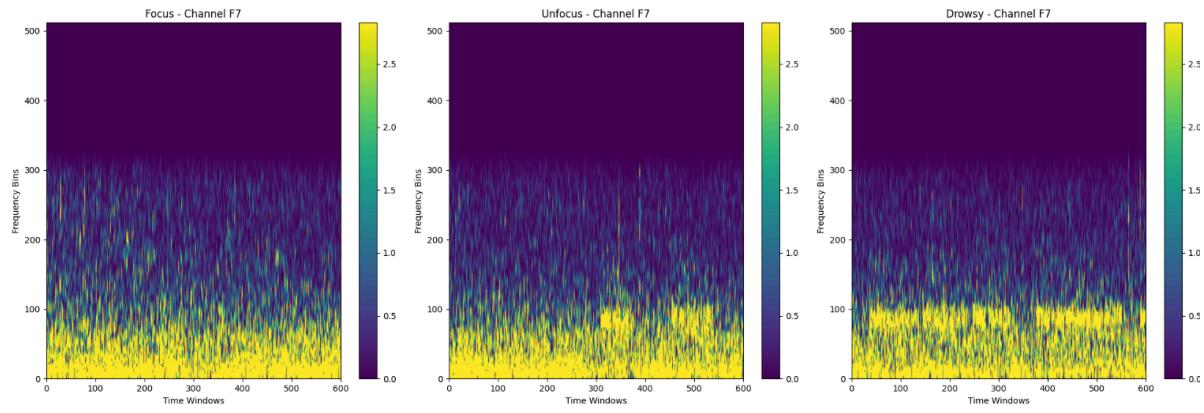
Tính toán năng lượng phổ của tín hiệu bằng cách lấy bình phương biên độ của STFT.

Các tham số như cửa sổ Blackman, tần số mẫu, độ dài cửa sổ và số điểm FFT được cấu hình để phân tích tín hiệu trong mỗi cửa sổ thời gian.

Công thức cửa sổ Blackman:

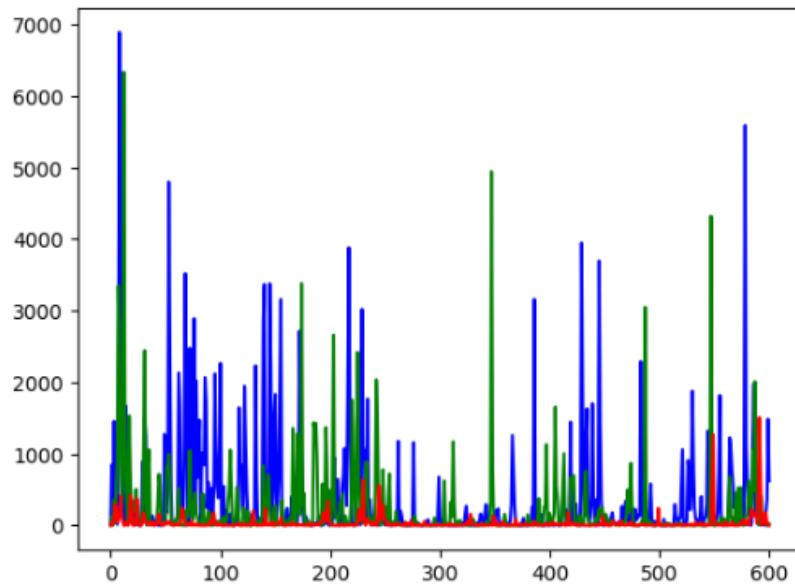
$$w(n) = 0.42 - 0.5 \cos\left(\frac{2\pi n}{M-1}\right) + 0.08 \cos\left(\frac{4\pi n}{M-1}\right), \quad 0 \leq n \leq M-1$$

Kết quả: Năng lượng phô cho mỗi đối tượng, mỗi trạng thái và mỗi ngày thí nghiệm, giúp phân tích các đặc trưng tần số của tín hiệu EEG trong từng khoảng thời gian.



Hình 8. So sánh phô năng lượng của Focus(bên trái), Unfocus(ở giữa) và Drowsy(bên phải)

Ở các trạng thái thì tần số sóng não của (tập trung > không tập trung > buôn ngủ) nhưng ở PSD thì ngược lại ta nhìn hình phân bố có thể thấy dải năng lượng vàng phân bố ở trạng thái buôn ngủ là nhiều nhất tiếp đó là không tập trung cuối cùng là tập trung => Tín hiệu PSD sẽ giảm ở các tần số cao và tăng ở các tần số thấp



Hình 9. mức năng lượng so sánh 3 trạng thái

Nhận xét: màu đỏ tượng trưng cho trạng thái buôn ngủ, màu xanh lá(không tập trung) cường độ mạnh nhưng mạnh nhất là màu xanh dương(tập trung)

5.2 Gộp bins tần số

Mục đích: Giảm số lượng bins trong dữ liệu, giúp tối ưu hóa việc phân tích và giảm độ phức tạp của mô hình.

Cách thức:

Duyệt qua từng trạng thái, đối tượng và ngày thí nghiệm

Gộp bins tần số trong mỗi kênh của dữ liệu EEG bằng cách tính trung bình các giá trị năng lượng trong mỗi khoảng bin_size

5.3 Tính trung bình qua cửa sổ chạy

Mục đích: Giúp cho các tín hiệu trở nên mượt mà hơn và dễ dàng phân tích hơn.

Cách thức:

Duyệt qua từng trạng thái, đối tượng và ngày thí nghiệm

Đối với mỗi cửa sổ thời gian có kích thước window_size, hàm sẽ tính trung bình các giá trị phổ trong cửa sổ đó dọc theo các kênh và bins tần số

5.4 Chuẩn hoá biến đổi Logarit

Mục đích: Làm cho các giá trị phổ tần số có thể dễ dàng so sánh và phân tích hơn.

Cách thức:

Khởi tạo từ điển ‘focus’, ‘unfocus’, ‘drowsy’ có kích thước (số kênh * số bins * số cửa sổ thời gian)

Dữ liệu phổ tần số từ từng trạng thái (focus, unfocus, drowsy) cho mỗi đối tượng và ngày được gộp lại và chuyển thành một mảng phẳng, với mỗi cột đại diện cho một cửa sổ thời gian.

Áp dụng phép biến đổi logarit lên từng mảng làm mượt các giá trị phổ tần số và đưa các giá trị về dạng dễ so sánh, đồng thời giảm độ rộng của giá trị dài động (dynamic range).

6 Label and Split Data

6.1 Khởi tạo nhãn

Ba nhãn ‘label_focus’, ‘label_unfocus’, ‘label_drowsy’ được khởi tạo cho mỗi trạng thái:

$$\text{focus} = 0, \text{unfocus} = 1, \text{drowsy} = 2$$

Các dữ liệu phô tần số từ ba trạng thái (focus, unfocus, drowsy) cho từng đối tượng và từng ngày được gộp lại trong một mảng. Sau đó, mảng của từng đối tượng và trạng thái được nối với nhau (axis=1), tạo thành một ma trận lớn hơn để chuẩn bị

Các nhãn tương ứng với từng trạng thái được thêm vào danh sách target trong quá trình lặp qua từng đối tượng và ngày. Các nhãn này sẽ được sử dụng trong bài toán phân loại để chỉ ra trạng thái tinh thần của đối tượng tại mỗi thời điểm.

Kết quả: Mảng subj chứa dữ liệu phô tần số đã được kết hợp từ tất cả các đối tượng và các ngày, với chiều dài (số mẫu) là tổng số mẫu từ tất cả các trạng thái và đối tượng. Mảng target chứa nhãn tương ứng cho mỗi mẫu trong dữ liệu.

6.2 Chia dữ liệu

Mục đích: Chuẩn bị dữ liệu cho mô hình học máy

Chia dữ liệu theo tỷ lệ 80:20 với dữ liệu để huấn luyện và nhãn để kiểm tra.

Chuẩn hóa dữ liệu huấn luyện, loại bỏ trung bình và chia cho độ lệch chuẩn.

7 Model theory

7.1 SVM

Support Vector Machine (SVM) là một thuật toán học có giám sát, được sử dụng trong các bài toán phân loại và hồi quy. Trong phân loại, SVM hoạt động bằng cách tìm kiếm một *siêu phẳng (hyperplane)* tối ưu để phân chia dữ liệu thành các lớp khác nhau. Mục tiêu của SVM là tối đa hóa khoảng cách giữa các điểm dữ liệu gần nhất của hai lớp và siêu phẳng phân chia chúng. Khoảng cách này được gọi là *lề (margin)*, và một siêu phẳng với lề lớn hơn thường có khả năng tổng quát hóa tốt hơn khi phân loại các điểm dữ liệu mới.

Nguyên lý hoạt động của SVM:

Tìm Siêu phẳng Tối ưu: SVM tìm kiếm siêu phẳng tối ưu để phân chia các lớp bằng cách tối đa hóa lề giữa các điểm dữ liệu gần nhất của hai lớp (các điểm này được gọi là các vector hỗ trợ (support vectors)). Đối với bài toán tuyến tính, siêu phẳng này có thể được biểu diễn bởi phương trình:

$$w^*x + b = 0$$

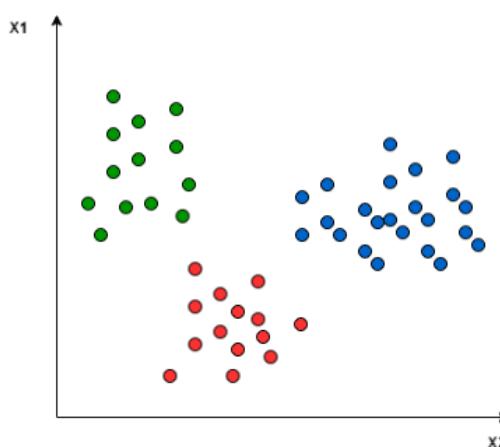
Trong đó:

w: Vector trọng số, hay *vector pháp tuyến (normal vector)* của siêu phẳng.

x: Vector đại diện cho các điểm dữ liệu.

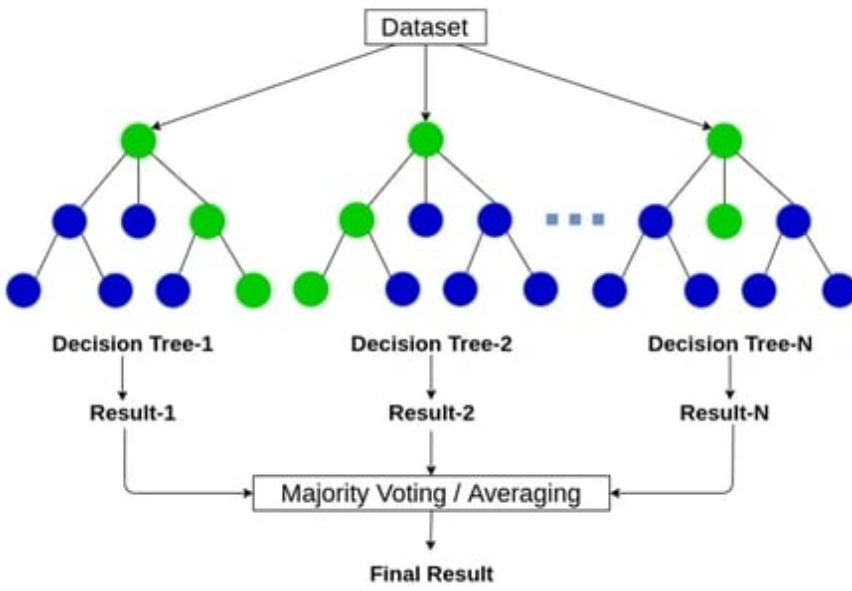
b: Bias, hay độ dịch của siêu phẳng.

Tối ưu hóa Lề: SVM tối ưu hóa lề bằng cách tìm các vector hỗ trợ và thiết lập khoảng cách giữa chúng và siêu phẳng tối ưu. Lề lớn hơn giúp giảm nguy cơ overfitting, cải thiện khả năng dự đoán trên dữ liệu mới.



7.2 Random Forest

Cụ thể thuật toán này tạo ra nhiều cây quyết định mà mỗi cây quyết định được huấn luyện dựa trên nhiều mẫu con khác nhau và kết quả dự báo là bầu cử (voting) từ toàn bộ những cây quyết định. Như vậy một kết quả dự báo được tổng hợp từ nhiều mô hình nên kết quả của chúng sẽ không bị chênh lệch. Đồng thời kết hợp kết quả dự báo từ nhiều mô hình sẽ có phương sai nhỏ hơn so với chỉ một mô hình. Điều này giúp cho mô hình khắc phục được hiện tượng quá khớp



Hình 10. Nguyên lý Random Forest

8 Model Built

8.1 SVM

Model Support Vector Machine được sử dụng trong bài toán này là mô hình phân lớp với kernel được sử dụng là “RBF” (Radial Basis Function). Tham số đi kèm là C=1.0 và set cho nó một random state để kết quả được cố định.

Hiệu suất tổng thể:

Độ chính xác trên tập huấn luyện: 96.40%.

Độ chính xác trên tập kiểm tra: 95.33%.

→ Mô hình có độ chính xác cao trên cả hai tập, cho thấy khả năng khai quát hóa tốt và không có dấu hiệu overfitting rõ rệt.

Lớp 0(focus) và 2(unfocus) có tỷ lệ dự đoán đúng rất cao trong confusion metrics. Lớp 1(unfocus) có số lượng nhầm lẫn lớn nhất, chủ yếu bị nhầm với lớp 0 (156 trường hợp). Điều này có thể phản ánh đặc điểm của lớp này tương tự với lớp 0(focus).

Precision: Trung bình đạt 0.95, cho thấy dự đoán của mô hình có độ chính xác cao.

Recall: Trung bình đạt 0.95, thể hiện mô hình nhận diện tốt các mẫu thực sự thuộc từng lớp.

F1-Score: Đều đạt 0.95 ở mức macro và weighted average, cho thấy sự cân bằng giữa precision và recall.

Thời gian huấn luyện và dự đoán ở mức lần lượt là 493 giây và 32 giây → Thời gian xử lý khá hợp lý đối với bài toán phân loại lớn.

8.2 Random Forest

Model áp dụng thứ hai mà chúng em áp dụng đó là model Random Forest, một trong những model cho khả năng phân loại rất tốt dựa vào cơ chế của Decision Tree.

Hiệu suất tổng thể:

Độ chính xác trên tập huấn luyện: xấp xỉ 100%.

Độ chính xác trên tập kiểm tra: 98,94%.

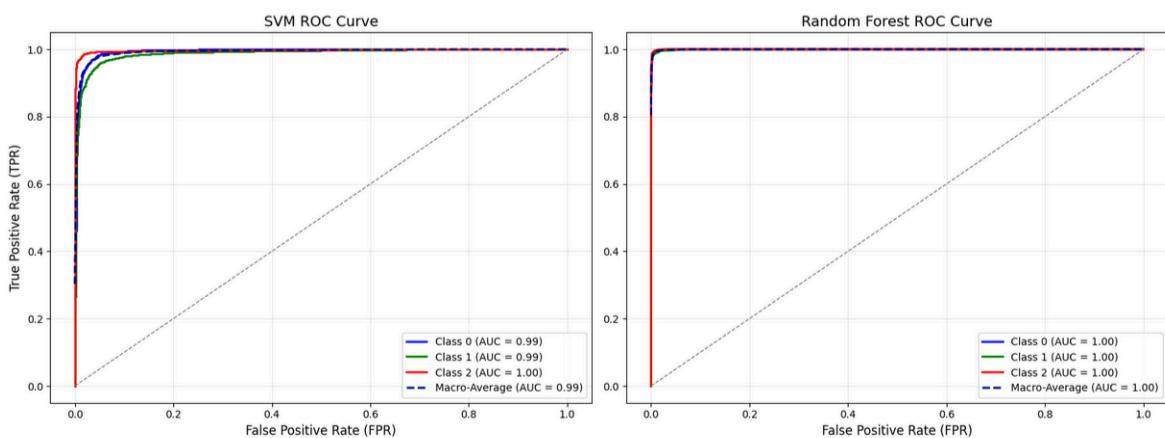
→ Mô hình đem lại kết quả rất tốt khi gần như tuyệt đối trên tập train và kết quả mang lại trên tập test cho thấy khả năng dự đoán chính xác các mẫu dữ liệu mới là rất cao.

Các chỉ số đánh giá khác (precision, recall, F1-score): Các chỉ số này đều đạt giá trị cao gần 1.0 cho từng lớp, cho thấy mô hình có khả năng phân loại chính xác các mẫu thuộc mỗi lớp.

Thời gian huấn luyện tương đối ngắn (29.79 giây) và thời gian dự đoán rất nhanh (0.10 giây), cho thấy mô hình có thể được áp dụng hiệu quả trong các hệ thống thực tế.

9 Performance Evaluation

9.1 ROC curves for model evaluation.



Model SVM: nhãn 2(drowsy) cho diện tích lớn nhất tiếp đến là nhãn 0(focus) và nhỏ nhất là nhãn 1(unfocus). Nhìn chung diện tích bên dưới đường cong (AUC) xấp xỉ 1 ở 3 nhãn, đường cong ROC sát bên trái điều này cho thấy mô hình có thể đạt được độ chính xác cao mà không làm giảm độ nhạy (sensitivity).

Model Random Forest: Tương tự như model SVM nhưng cho kết quả tốt hơn ở cả 3 nhãn. Cả AUC và đường cong ROC đều có kết quả gần như chính xác tuyệt đối ở tập test.

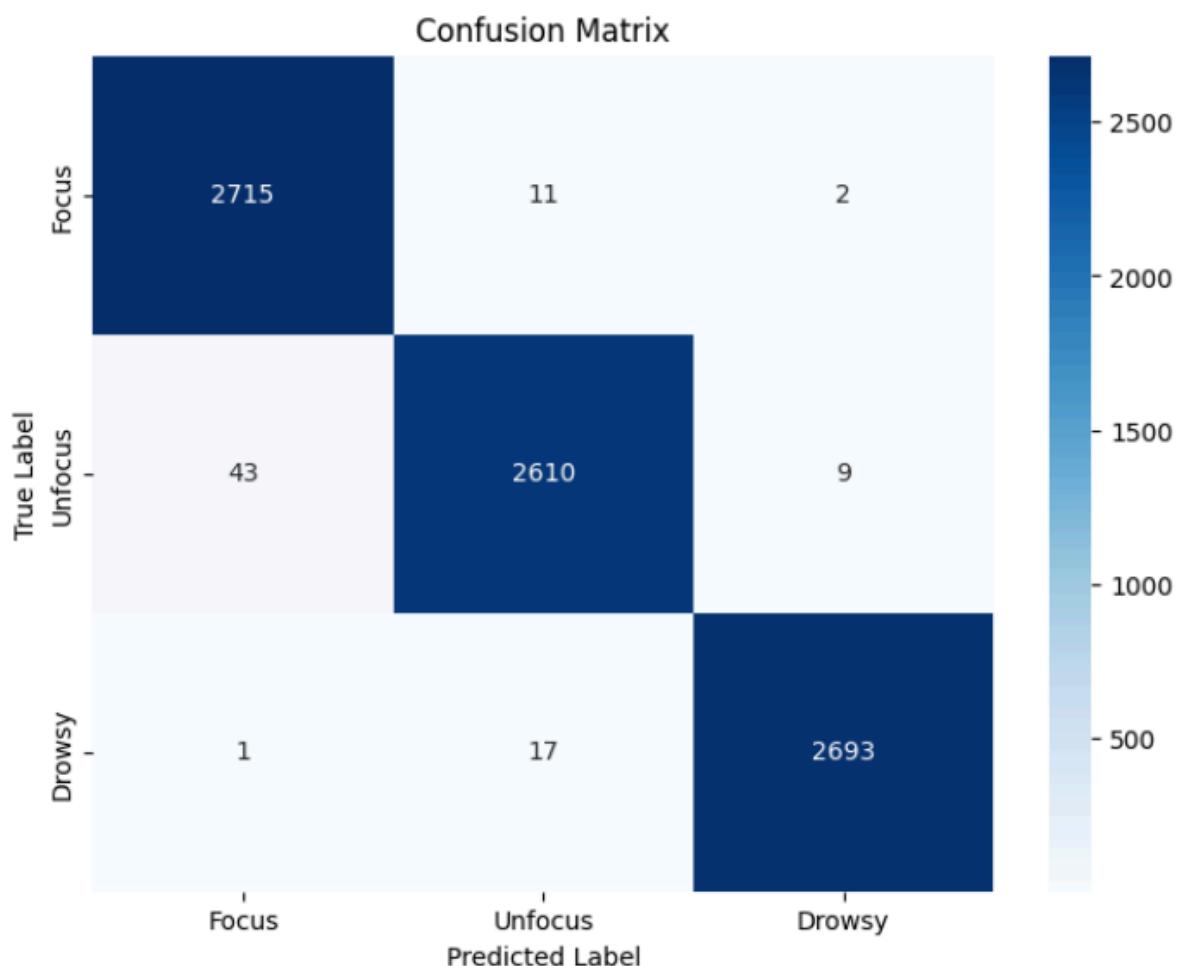
→ Cả hai mô hình đều cho hiệu năng rất cao và đều có thể sử dụng để dự đoán mà cho ra kết quả có sai lệch rất thấp.

9.2 Model Selection

| | Model | Train Accuracy | Test Accuracy | Training Time (seconds) | Prediction Time (seconds) |
|---|---------------|----------------|---------------|-------------------------|---------------------------|
| 0 | SVM | 0.963953 | 0.953339 | 493.749210 | 32.474164 |
| 1 | Random Forest | 1.000000 | 0.989384 | 29.786671 | 0.101877 |

Dựa trên kết quả thì model Random Forest là lựa chọn tốt hơn do thời gian train và predict đều rất ổn tương đồng thời Accuracy cũng cao hơn SVM.

9.3 Confusion matrices



Kết quả cho ra rất cao nên tiềm ẩn khả năng bị Leakage data nên khả năng là dữ liệu đã gấp vấn đề trong quá trình xử lí. Trên cả 2 mô hình cho kết quả đều cao trên cả train và test. Chúng em cần xem xét lại để xử lí một cách hợp lý vấn đề gấp phải.

10 Hướng giải quyết cho vấn đề dự đoán của các model ra quá cao

Các bước xử lý dữ liệu bằng STFT:

1. Tính phổ năng lượng (power spectrum) của tín hiệu EEG theo tần số bằng cách sử dụng STFT (Short-Time Fourier Transform).
2. Gộp các tần số thành các bins có kích thước khoảng 0.5Hz để giảm kích thước dữ liệu và tập trung vào các dải tần số quan trọng từ 0.5 - 18Hz
3. Tính trung bình qua các cửa sổ thời gian chạy (sliding window) 15s để làm mượt tín hiệu và loại bỏ nhiễu trong dữ liệu.

-> Quá trình trích xuất đặc trưng bên dưới ở bước 3 chúng em đã sử dụng sliding window với thời gian 15s để tính trung bình các giá trị trích xuất ở miền tần số với cửa sổ trượt 15s. Chính vì thế ở bước chia dữ liệu của tụi em được chia thành tập train và test một cách ngẫu nhiên giữa các đối tượng gây nên vấn đề Leakage Data.

Chúng em đã nghĩ ra hướng giải quyết bằng cách sử dụng cách chia dữ liệu với tập test là dữ liệu của từng người tham gia (subject) để đánh giá khách quan nhất về hiệu suất của mô hình.

Tiến hành chia dữ liệu lựa chọn Subject 5 làm tập test và 4 Subject còn lại làm tập train.

10.1 Thử nghiệm trên mô hình SVM

```

Train Accuracy of SVM model: 0.9140141665919483
Test Accuracy of SVM model: 0.604344122657581

Confusion Matrix:
[[1904  354   90]
 [1139  713  496]
 [ 433  275 1640]]

Classification Report:
      precision    recall  f1-score   support
          0       0.55     0.81     0.65     2348
          1       0.53     0.30     0.39     2348
          2       0.74     0.70     0.72     2348

      accuracy                           0.60     7044
     macro avg       0.61     0.60     0.59     7044
  weighted avg       0.61     0.60     0.59     7044
  
```

10.2 Thử nghiệm trên mô hình Random Forest

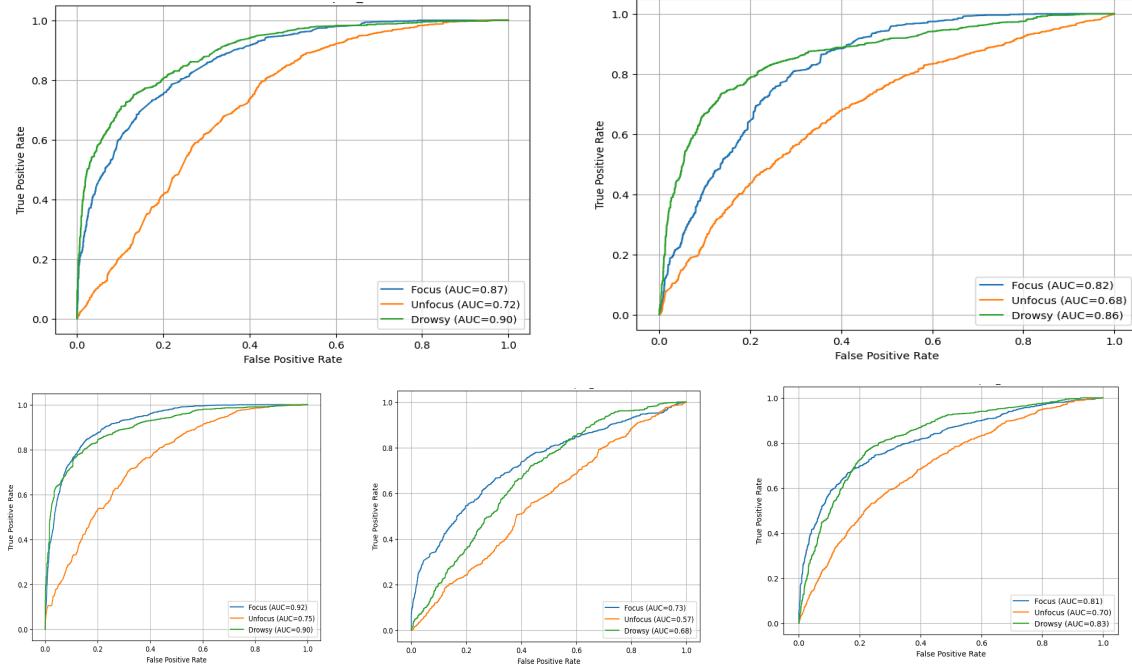
```

Train Accuracy of Random Forest model: 0.651005708479034
Test Accuracy of Random Forest model: 0.630465644520159
Confusion Matrix:
[[1418 478 452]
 [ 394 1255 699]
 [ 115 465 1768]]

```

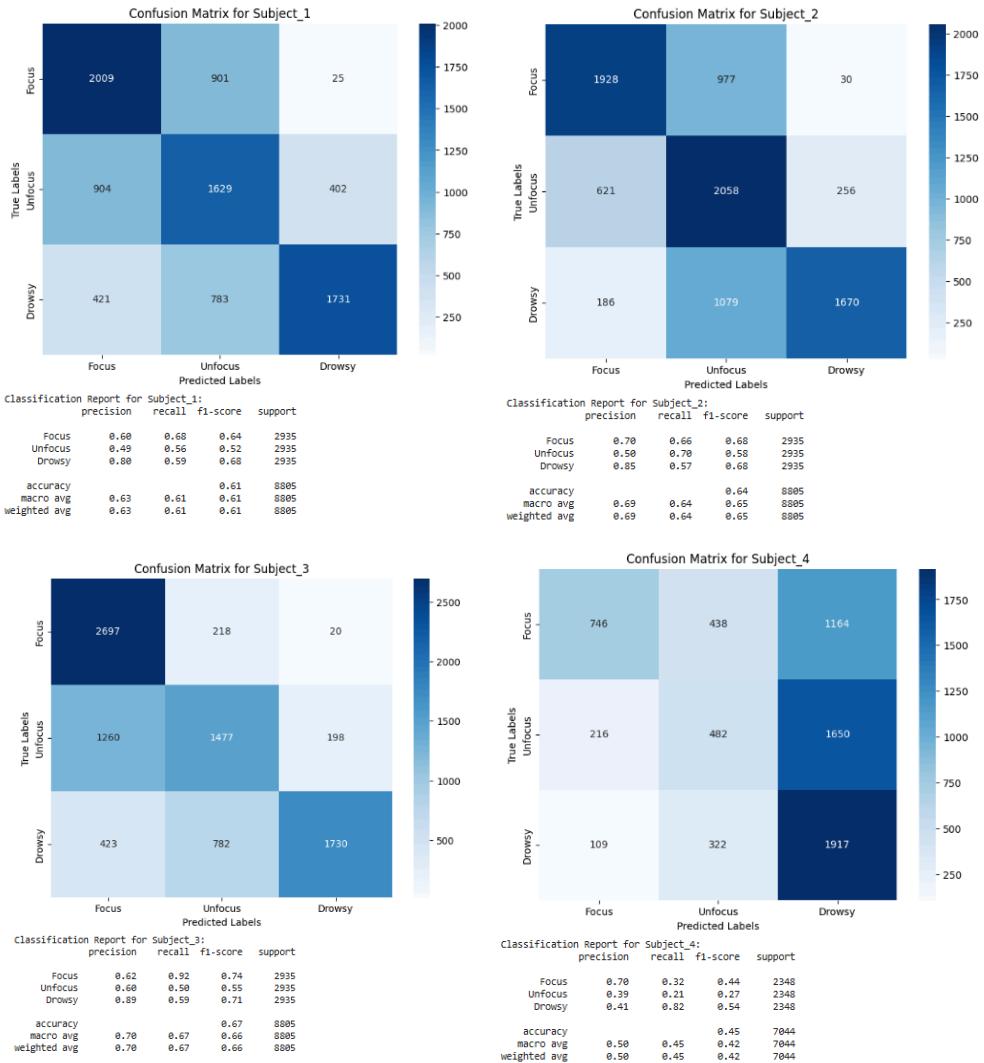
| Classification Report: | | precision | recall | f1-score | support |
|------------------------|---|-----------|--------|----------|---------|
| | | 0.74 | 0.60 | 0.66 | 2348 |
| | 0 | 0.57 | 0.53 | 0.55 | 2348 |
| | 1 | 0.61 | 0.75 | 0.67 | 2348 |
| accuracy | | | | 0.63 | 7044 |
| macro avg | | 0.64 | 0.63 | 0.63 | 7044 |
| weighted avg | | 0.64 | 0.63 | 0.63 | 7044 |

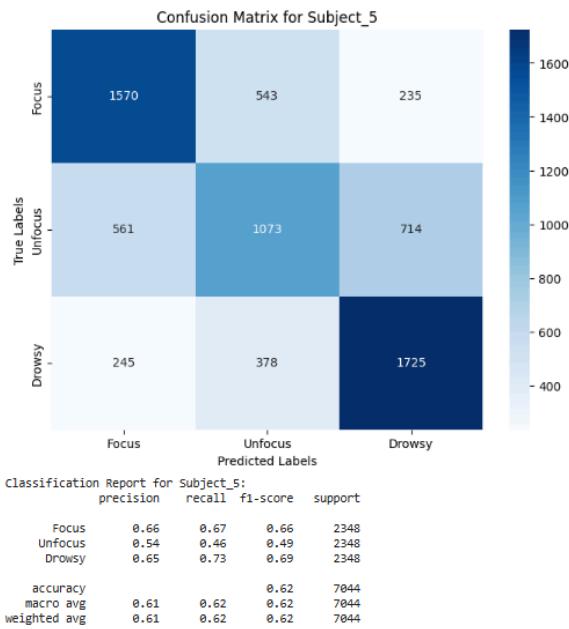
10.3 ROC curves for model evaluation



Nhận xét: Nhìn chung từ đường cong ROC thì có thể thấy được các nhãn của Focus và Drowsy cho kết quả tốt hơn ở trên 5 Fold dữ liệu đánh giá khi các đường cong sát với đường viền hơn có thể thấy được rằng 2 nhãn này có khả năng phân loại tốt hơn so với nhãn Unfocus.

10.4 Confusion matrices

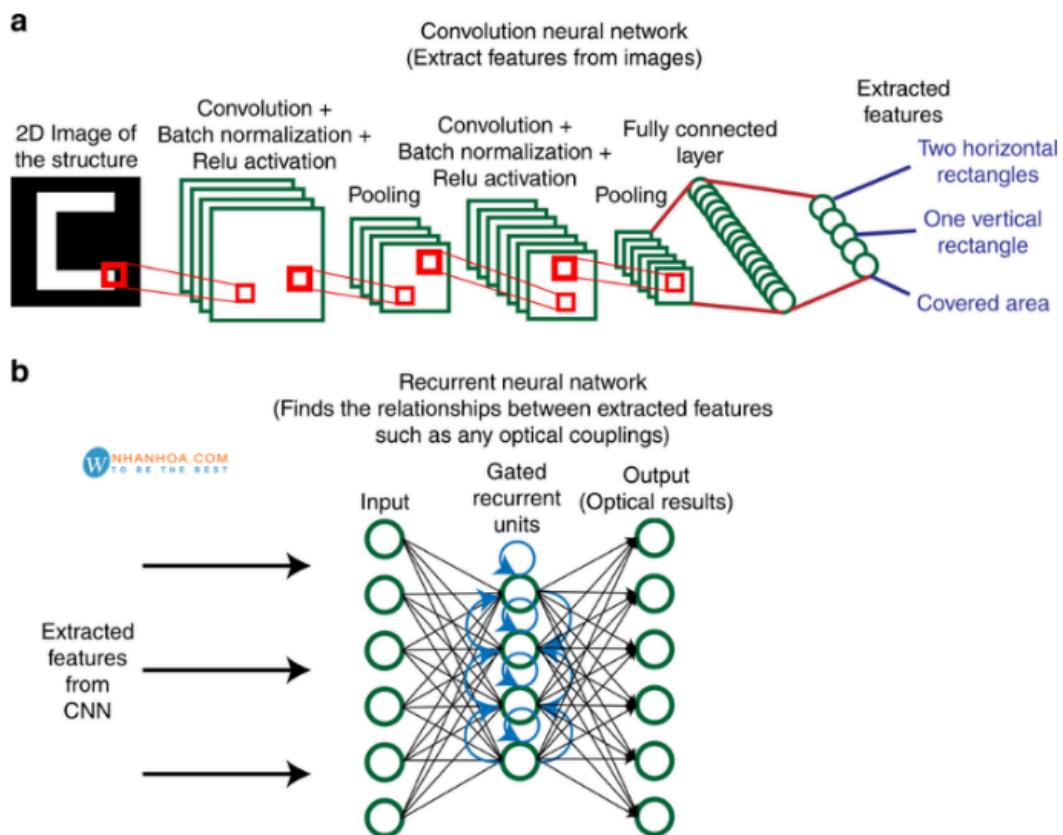




Từ các biểu đồ Confusion matrix với 5 Subjects chúng em thấy rằng ở trường hợp Subject 4 làm tập test thì đã phần dữ liệu sẽ dự đoán về lớp Drowsy có thể dữ liệu của người thứ 4 này có tín hiệu sóng ở dải băng tần thấp hoạt động mạnh hơn so với các người tham gia khác dẫn đến dữ liệu có sự sai lệch rõ ràng khi dự đoán trên tập test là Subject 4.

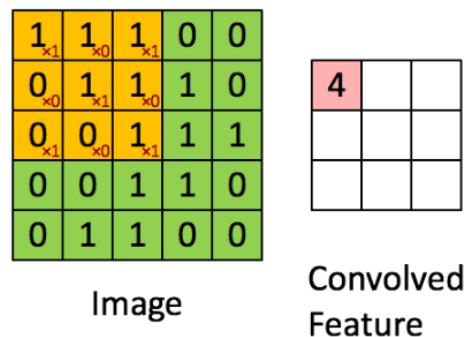
11 CNN (Đè xuất mở rộng)

Convolutional Neural Network hay còn được viết tắt là CNN, trong tiếng Việt được gọi là mạng nơ-ron tích chập, là một trong những mô hình Deep Learning tiên tiến và hiện đại nhất hiện nay. Nhờ có CNN mà chúng ta có thể dễ dàng tạo dựng hệ thống thông minh và có độ chính xác cao.



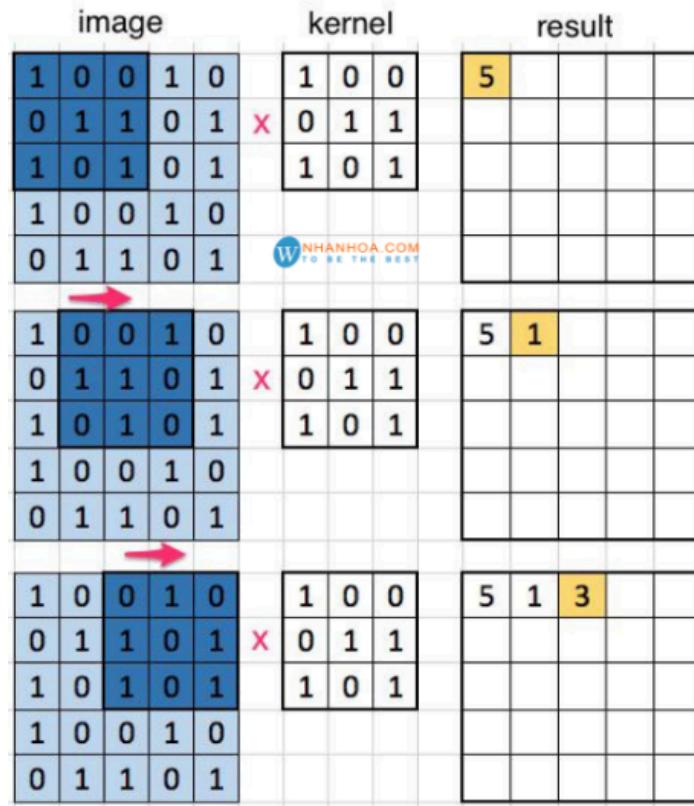
Convolutional Neural Network được sử dụng nhiều trong các bài toán nhận dạng các object trong ảnh. Để tìm hiểu tại sao thuật toán này được sử dụng rộng rãi cho việc nhận dạng (detection), chúng ta hãy cùng tìm hiểu về thuật toán này.

Convolutional là gì? Là một cửa sổ trượt (Sliding Windows) trên một ma trận mô tả như hình



Cấu trúc của Convolutional Neural Network Mạng CNN gồm nhiều lớp Convolution chồng lên nhau, sử dụng các hàm và tanh để kích hoạt các trọng số. Mỗi một lớp sau khi được kích

hoạt sẽ cho ra kết quả trùu tương cho các lớp tiếp theo. Mỗi layer kế tiếp chính là thể hiện kết quả của layer trước đó.

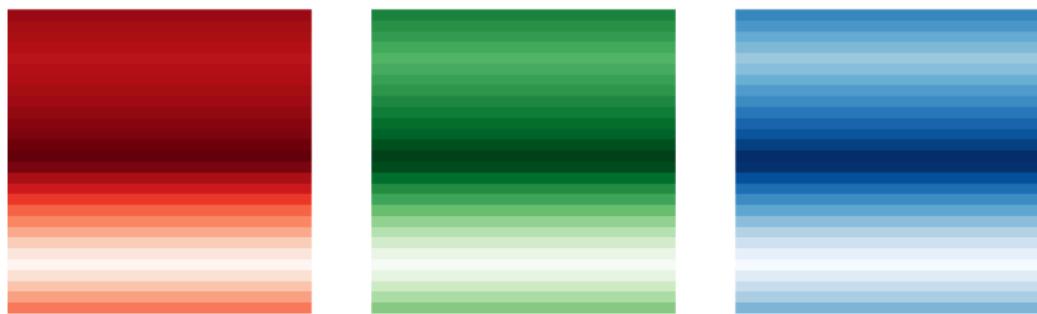


Cách xây dựng mô hình

Trích xuất dữ liệu thành 3 band tần số

{“Theta”: (4, 8), “Alpha”: (8, 13), “Beta”: (13, 18) }

Chuyển dữ liệu sau khi trích xuất thành ảnh, mỗi tấm ảnh đại diện cho 1 band tần số



Hình 13. Các ảnh tương ứng với band tần số Red(Theta), Green(Alpha) và Blue(Beta)
Chuẩn bị và gán dữ liệu cho toàn bộ dữ liệu sau khi trích xuất, sử dụng mô hình CNN, sau khi train thu được

```
Epoch 1/20, Loss: 1.0990, Accuracy: 32.82%
Epoch 2/20, Loss: 1.0988, Accuracy: 33.14%
Epoch 3/20, Loss: 1.0987, Accuracy: 33.28%
Epoch 4/20, Loss: 1.0987, Accuracy: 33.18%
Epoch 5/20, Loss: 1.0986, Accuracy: 33.77%
Epoch 6/20, Loss: 1.0987, Accuracy: 33.17%
Epoch 7/20, Loss: 1.0987, Accuracy: 33.39%
Epoch 8/20, Loss: 1.0987, Accuracy: 32.93%
Epoch 9/20, Loss: 1.0987, Accuracy: 33.09%
Epoch 10/20, Loss: 1.0987, Accuracy: 33.29%
Epoch 11/20, Loss: 1.0987, Accuracy: 33.06%
Epoch 12/20, Loss: 1.0987, Accuracy: 32.99%
Epoch 13/20, Loss: 1.0987, Accuracy: 33.73%
Epoch 14/20, Loss: 1.0987, Accuracy: 33.02%
Epoch 15/20, Loss: 1.0987, Accuracy: 32.89%
Epoch 16/20, Loss: 1.0987, Accuracy: 32.89%
Epoch 17/20, Loss: 1.0987, Accuracy: 33.09%
Epoch 18/20, Loss: 1.0987, Accuracy: 33.15%
Epoch 19/20, Loss: 1.0987, Accuracy: 33.23%
Epoch 20/20, Loss: 1.0986, Accuracy: 33.39%
```

Việc phát triển CNN 2D cho dữ liệu EEG là hướng đi tiềm năng vì khả năng khai thác đặc trưng không gian khi dữ liệu được chuyển đổi phù hợp. EEG chứa thông tin quan trọng cả ở miền thời gian và tần số, nên việc chuyển đổi tín hiệu thành bản đồ thời gian-tần số (spectrogram) giúp CNN 2D học đặc trưng tốt hơn. Ngoài ra, nếu dữ liệu được sắp xếp lại dựa trên vị trí các điện cực EEG, CNN 2D có thể khai thác mối quan hệ không gian giữa các kênh. CNN 2D cũng nổi bật trong việc học các đặc trưng cục bộ thông qua các kernel và có thể mở rộng hiệu quả khi kết hợp với mô hình LSTM để học cả đặc trưng thời gian và không gian. Sự thành công của CNN 2D trong phân tích hình ảnh và dữ liệu thời gian-tần số là cơ sở cho việc ứng dụng vào dữ liệu EEG khi được xử lý đúng cách.