MTCAIC3 COMPETITION

DETAILED REPORT ON EEG CLASSIFICATION CODE

Team Name	**Ain El3aql**
Team members	• Eman Ashraf Ebrahim Elsayed • Raheeq Ayman Ramadan • Shrouk Mohamed Aboalela • Mohamed Khaled Gamal • Abdelrahman Ahmed Taher

Introduction

As a participating team in the **AIC-3**: Egypt National Artificial Intelligence Competition, we were excited to take on the challenge of building robust AI models for Brain-Computer Interface (**BCI**) tasks. This competition gave us the opportunity to dive deep into two major EEG-based paradigms: Steady-State Visual Evoked Potentials (**SSVEP**) and Motor Imagery (**MI**).

Throughout the competition, our focus was on developing accurate and efficient classification models capable of handling multi-channel EEG signals. We explored a variety of signal processing techniques and machine learning approaches, combining traditional methods like Common Spatial Patterns (CSP) and Fast Fourier Transform (FFT) with deep learning architectures such as **EEGNet** and **DeepConvNet**.

To optimize our performance, we implemented multiple preprocessing pipelines, experimented with different feature extraction strategies, and carefully tuned our models using cross-validation and held-out validation sets. We also built ensemble models to combine the strengths of different approaches and improve generalization.

This report outlines the steps we took during the competition, including **data preprocessing**, **model design**, **training strategies**, and **evaluation results**. It also discusses the challenges we encountered, how we addressed them, and the insights we gained throughout the project.

Our participation in **AIC-3** was not only a technical challenge but also a valuable learning experience. We're proud of the progress we've made and the solutions we've built, and we're excited to share the outcomes in the sections that follow.

Motor Imagery (MI) Task

1. Overview

In the Motor Imagery (**MI**) task, we aimed to classify EEG signals recorded during imagined hand movements. Our pipeline begins with preprocessing steps including EEG channel selection, band-pass filtering (8–30 Hz), and epoch segmentation. We then extract a diverse set of features using multiple techniques such as Filter Bank Common Spatial Pattern (FBCSP), Independent Component Analysis (ICA), Higuchi Fractal Dimension (HFD), Auto-Regressive (AR) modeling, Short-Time Fourier Transform (STFT), and Matching Pursuit (MP).

These features are fed into various classifiers including traditional models like Linear Discriminant Analysis (LDA) and Support Vector Machines (SVM), as well as deep learning models such as EEGNet, DeepConvNet, BiLSTM, and CNN-LSTM with attention. The pipeline is modular, allowing experimentation with different feature-model combinations. The best results were achieved using FBCSP features with EEGNet, demonstrating high accuracy in distinguishing between different motor imagery classes.

---

2. System Architecture

2.1 Overview

The system is designed to process and classify EEG signals for motor imagery tasks, following a structured pipeline:

1. **Data Loading**: Reads and organizes EEG data from CSV files, filtering for MI tasks and encoding labels.
2. **Preprocessing**: Applies channel selection, bandpass filtering, epoch extraction, and normalization to prepare clean EEG epochs.
3. **Feature Extraction**: Uses FBCSP to extract discriminative spatial patterns across multiple frequency bands.

4. **Classification**: Employs EEGNet, a lightweight convolutional neural network, for accurate classification.

5. **Evaluation and Output**: Generates predictions and confidence scores, saving them in the required format.

The pipeline is modular, allowing experimentation with alternative feature extraction methods (e.g., CSP, STFT, ICA) and classifiers (e.g., LDA, SVM, BiLSTM), though we focuse on the **FBCSP + EEGNet** configuration specified in the code's example execution.

---

3. Employed Methodologies

3.1 Data Loading

- **Implementation**:
  - Reads index CSV files (train.csv, validation.csv, test.csv) using Pandas.
  - Filters for MI tasks and applies **LabelEncoder** to transform categorical labels (e.g., "left_hand", "right_hand") into numerical format.
  - Returns DataFrames for each split and the fitted encoder for consistent label mapping.
- **Techniques**:
  - Pandas for data manipulation.
  - Scikit-learn's LabelEncoder for label preprocessing.

3.2 EEG Preprocessing (EEGPreprocessor)

- **Purpose**: Prepares raw EEG signals for feature extraction by removing noise and extracting standardized epochs.
- **Parameters**:
  - filter_low: **8 Hz** (alpha band).
  - filter_high: **30 Hz** (beta band).
  - SELECTED_CHANNELS: [**'FZ'**, **'C3'**, **'CZ'**, **'C4'**, **'PZ'**, **'PO7'**, **'OZ'**, **'PO8'**].

- SAMPLING_RATE: **250 Hz**.
- TRIAL_LENGTH: **2250** samples.

- **Steps**:
  - **Channel Selection**: Extracts data from 8 channels relevant to motor imagery, reducing noise from irrelevant electrodes.
  - **Bandpass Filtering**: Applies a 5th-order Butterworth filter (8–30 Hz) to isolate alpha and beta bands, which are critical for MI tasks due to their association with motor-related neural activity.
  - **Epoch Extraction**: Segments data into trials of 2250 samples, corresponding to individual MI tasks.
  - **Normalization**: Applies per-channel normalization (zero mean, unit variance) to handle inter-channel variability and stabilize feature distributions.
- **Output**: Epochs of shape (n_trials, 2250, 8).
- **Techniques**:
  - Butterworth filter using SciPy's butter and filtfilt for zero-phase filtering.
  - Per-channel normalization to mitigate amplitude variations.
- **Significance**: Preprocessing enhances signal quality, removes artifacts, and ensures consistent input for feature extraction, aligning with the competition's emphasis on robust signal processing.

3.3 Feature Extraction (FBCSPFeatures)

- **Purpose**: Extracts discriminative spatial patterns across multiple frequency bands using Filter Bank Common Spatial Patterns (FBCSP), a state-of-the-art method for MI tasks.
- **Parameters**:
  - n_components: 4 (number of CSP components per frequency band).

- freq_bands: [(8,12), (12,16), (16,24), (24,30)] (covering alpha and beta sub-bands).

- **Implementation**:
    - Applies bandpass filtering to each frequency band using a 5th-order Butterworth filter.
    - Fits a CSP model for each band using MNE-Python's CSP class, maximizing variance between MI classes.
    - Concatenates CSP features from all bands to form a feature vector of shape (n_trials, n_components * n_bands).
- **Techniques**:
    - Butterworth filtering for precise frequency band isolation.
    - CSP with logarithmic variance to enhance discriminative power.
- **Significance**: FBCSP extends traditional CSP by processing multiple frequency bands, capturing richer spatial and spectral patterns, which is critical for achieving high accuracy in MI classification.

  We also evaluated classic CSP (single band) and ICA features, but FBCSP consistently yielded better validation accuracy.

3.4 Classification (EEGNet)

- **Purpose**: Classifies FBCSP features into MI categories using EEGNet, a compact convolutional neural network designed for EEG data.
- **Architecture**:
    - **Temporal Convolution**: Conv2d layer (16 filters, kernel size (1,64), padding (0,32)) to capture temporal patterns in EEG signals.
    - **Depthwise Convolution**: Conv2d layer (32 filters, kernel size (8,1), groups=16) to model spatial relationships across channels.

- **Batch Normalization and ELU**: Stabilizes training and introduces non-linearity.
- **Average Pooling**: Reduces dimensionality (kernel size (1,4)).
- **Dropout**: Regularizes the model (dropout rate: 0.25).
- **Fully Connected Layer**: Maps to num_classes outputs for classification.

- **Input Shape**: (batch, 1, channels, samples) (reshaped from preprocessed epochs: (n_trials, 8, 2250)).
- **Training**:
  - Optimizer: Adam (learning rate: 1e-3).
  - Loss: CrossEntropyLoss.
  - Epochs: 20 (configurable).
  - Batch Size: 32 (configurable).
  - Saves best model checkpoint (best_EEGNet.pt) based on validation accuracy.
- **Techniques**:
  - PyTorch for model implementation and training.
  - DataLoader for efficient batch processing.
  - Softmax for generating confidence scores.
- **Significance**: EEGNet's lightweight architecture is optimized for EEG data, offering high accuracy with low computational cost, making it suitable for BCI applications and the competition's real-time evaluation criteria.

3.5 Model Training (ModelTrainer)

- **Purpose**: Provides a unified interface for training EEGNet and generating test predictions.
- **Implementation**:
  - Initializes EEGNet with the appropriate input shape ((8, 2250)) and number of classes (determined by LabelEncoder).

- Uses PyTorch's training loop with Adam optimizer and CrossEntropyLoss.
- Evaluates validation accuracy after each epoch and saves the best model checkpoint.
- Generates predictions and confidence scores for the test set using predict.
- The confidence scores were further analyzed to understand model certainty and assist in ensemble or post-processing (if applicable). Additionally, we experimented with combining outputs from multiple models by selecting the label with the highest confidence score across models to improve overall prediction accuracy.

- **Output**: A DataFrame with columns id, label, and confidence, saved as vs_submission.csv.
- **Significance**: The unified trainer ensures consistent training and evaluation, supporting the competition's reproducibility requirements.

---

4. Technical Details

4.1 Code Organization and Clarity

- **Modularity**: The code is structured into reusable classes:
  - **EEGPreprocessor**: Handles preprocessing.
  - **FBCSPFeatures**: Manages feature extraction.
  - **EEGNet**: Defines the classifier.
  - **ModelTrainer**: Orchestrates training and inference.
- **Conventions**: Uses scikit-learn's BaseEstimator and TransformerMixin for preprocessing and feature extraction, and PyTorch's nn.Module for EEGNet.
- **Documentation**: Includes detailed docstrings for all classes and functions, explaining inputs, outputs, and functionality.
- **File Structure**:

  - Main script (main_MI) orchestrates the pipeline.
  - Output saved as vs_submission.csv.
  - Model checkpoint saved as best_EEGNet.pt.
- **Significance**: The clear, modular design facilitates maintenance, debugging, and experimentation, aligning with the competition's emphasis on code clarity.

---

5. Evaluation and Results

5.1 Primary Metric: Mean Classification Accuracy

- **Evaluation Setup**: The system generates predictions for the held-out test set, saved in vs_submission.csv with columns id, label, and confidence. Accuracy is computed for MI trials and contributes to the competition's mean accuracy metric (averaged with SSVEP performance).
- **Expected Performance**: The FBCSP+EEGNet combination is expected to achieve high accuracy due to FBCSP's discriminative feature extraction and EEGNet's robust classification. Validation accuracy is reported during training to guide model selection, with the best model checkpoint used for test predictions.
- **Output Format**: id,label,confidence

  The output file generated by this code with id, label, and confidence columns, is processed by a secondary script that decodes the predictions and removes the confidence column. The resulting file conforms to the AIC-3 competition's required format, consisting of two columns: id and label.

---

8. Conclusion

The Motor Imagery (MI) pipeline we developed for the AIC-3 competition demonstrated strong performance through a combination of robust preprocessing, effective feature extraction using FBCSP, and the lightweight yet powerful EEGNet classifier. Our modular architecture allowed extensive experimentation with various components, ultimately identifying the FBCSP + EEGNet pairing as the most effective for this task. The system achieved high validation accuracy while maintaining computational efficiency and reproducibility, making it suitable for real-time BCI applications.

This phase of the competition highlighted the importance of selecting meaningful frequency bands, reducing noise through spatial filtering, and designing models tailored to EEG signals. Our work serves as a solid foundation for future improvements, including cross-subject generalization, transfer learning, or real-time deployment.

---

9. Challenges and Iterative Improvements

The development of the EEG classification system for the AIC-3 Motor Imagery (MI) task faced several challenges, which were addressed through targeted strategies to ensure high accuracy, robustness, and efficiency.

1. **Challenge: High Variability in EEG Signals** EEG signals vary across subjects and sessions due to noise, artifacts, and physiological differences, complicating classification.
    - **Solution**: The EEGPreprocessor applied channel selection (8 MI-relevant channels: FZ, C3, CZ, C4, PZ, PO7, OZ, PO8), 5th-order Butterworth bandpass filtering (8–30 Hz), per-channel normalization, and epoch extraction (2250 samples) to reduce noise and standardize data.
2. **Challenge: Extracting Discriminative Features** MI tasks produce subtle EEG patterns with low signal-to-noise ratios, requiring effective feature extraction.
    - **Solution**: The FBCSPFeatures class used Filter Bank Common Spatial Patterns (FBCSP) to extract spatial patterns across multiple frequency bands (8–12, 12–16, 16–24, 24–30 Hz), achieving the best validation accuracy. Alternatives like CSP, STFT, ICA, AutoRegression, Higuchi Fractal Dimension, and Matching Pursuit were implemented for flexibility.

3. **Challenge: Balancing Model Complexity and Efficiency** High accuracy is needed without excessive computational cost, especially for potential real-time applications.
   - **Solution**: The EEGNet model, a lightweight CNN with temporal and depthwise convolutions, was selected for its efficiency and high performance. Alternatives included LDA, SVM, Random Forest, DeepConvNet, BiLSTM, and CNN-LSTM for experimentation.
4. **Challenge: Reproducibility and Submission Format** The competition requires reproducible results and a specific output format (id, label).
   - **Solution**: The ModelTrainer saved model checkpoints (best_EEGNet.pt), and the output file vs_submission.csv (with id, label, confidence) was processed by a secondary script to remove the confidence column, ensuring compliance with the required format.

Best-Performing Configuration: The FBCSP+EEGNet pipeline delivered the highest validation accuracy due to FBCSP's discriminative feature extraction and EEGNet's efficient classification, making it the focus of the detailed pipeline explanation below.

Steady-State Visual Evoked Potentials (SSVEP) Task

Overview

In the Steady-State Visual Evoked Potential (SSVEP) task, the objective is to classify EEG responses elicited by flickering visual stimuli at predefined frequencies (7, 8, 10, 13 Hz). These stimuli generate characteristic frequency components in the EEG signal, enabling frequency-based classification.

We experimented with various preprocessing techniques, feature extraction methods (including Canonical Correlation Analysis (CCA), Filter Bank CCA (FBCCA), STFT, and wavelet features), and classification models such as traditional machine learning classifiers (e.g., LDA, SVM), in addition to deep learning architectures including CNNs, LSTMs, EEGNet, Transformers, and hybrid models like Transformer-LSTM.

The best results were achieved using **FBCCA features with a Transformer-LSTM hybrid classifier**, combined with **SelectKBest feature selection** and **StandardScaler normalization**. The pipeline is modular and supports easy experimentation across components.

---

2. System Architecture

2.1 Overview

The system consists of the following components for SSVEP classification:

1. **Data Loading:** Reads EEG metadata (train/val/test CSVs), filters SSVEP tasks, encodes labels.
2. **Preprocessing:** Applies bandpass filtering across multiple sub-bands (filter bank) and epoch extraction.
3. **Feature Extraction:** Employs **FBCCA**, which enhances canonical correlation analysis by applying multiple sub-band filters.

4. **Feature Selection & Normalization:** Uses StandardScaler and SelectKBest to reduce dimensionality and normalize features.
5. **Classification:** Trains deep learning models (e.g., Transformer-LSTM hybrid) on extracted features.
6. **Evaluation:** Generates predictions with confidence scores and exports results in submission format.

---

3. Employed Methodologies

3.1 Data Loading

- **Purpose:** Read CSVs (train.csv, validation.csv, test.csv) and encode categorical labels for SSVEP.
- **Techniques:**
    - pandas for data handling.
    - LabelEncoder from scikit-learn for transforming string labels to numerical.

3.2 EEG Preprocessing & Epoch Extraction

- **Purpose:** Extract EEG epochs and prepare signals for FBCCA.
- **Techniques:**
    - Filter bank creation using multiple bandpass filters:
        - Subbands: [(5–40), (6–38), (7–36), (8–34), (9–32)]
    - Filter design using a 4th-order Butterworth filter.
    - Epoch slicing per trial: 1750 samples @ 250Hz sampling rate (≈7 seconds).

3.3 Feature Extraction (FBCCA)

- **Purpose:** Extract correlation-based frequency domain features using **Filter Bank Canonical Correlation Analysis**.
- **Implementation:**

- For each sub-band, apply bandpass filtering and compute CCA correlation with reference sine/cosine signals for each target frequency.
- Weight each sub-band correlation using inverse sub-band index weights.

- **Advantages:**
  - Leverages frequency-specific brain responses.
  - Improves classification in presence of noise via sub-band averaging.

3.4 Feature Normalization and Selection

- **StandardScaler:** Normalize features to zero-mean and unit variance.
- **SelectKBest (f_classif):** Selects the top-k most statistically significant features.
  - Typically, 120–500 features were retained based on experiment.
- **Purpose:** Improves model generalization and reduces overfitting.

3.5 Classification Models

We tested multiple models for classifying the extracted FBCCA features:

Model Name	Description
EEGNet	Shallow CNN with temporal & spatial convolutions, works well with EEG
EnhancedFeatureClassifier	MLP with attention module for feature weighting
CNNClassifier	Conv1D layers + max pooling

Model Name	Description
SSVEPFormer	Transformer encoder with positional embeddings
BiLSTMClassifier	Bi-directional LSTM for temporal modeling
TransformerLSTMHybrid	Best-performing model combining transformer encoding and LSTM layers

- **Best Results:** TransformerLSTMHybrid achieved the highest validation accuracy and confidence.

---

4. Technical Details

4.1 Training Setup

- **Optimizer:** AdamW with cosine annealing learning rate scheduler.
- **Loss:** CrossEntropyLoss.
- **Regularization:** L2 penalty and dropout in all layers.
- **Early Stopping:** Patience = 15.
- **Best checkpoint saved:** 'best_SSVEP_model.pth'.

4.2 Input Pipeline

- Data loaded via FeatureDataset and wrapped in DataLoader.
- Batch size = 32; Training for up to 60 epochs.

---

5. Evaluation and Results

5.1 Metric

- **Primary:** Mean classification accuracy on the validation set.
- **Secondary:** Confidence scores using softmax output.

5.2 Output

- **Columns:** id, label, confidence.
- Final predictions saved in submission.csv.

5.3 Performance

- **Validation accuracy (best run):** Achieved up to 91%+
- **Confidence:** Mean confidence per prediction typically >0.94
- Compared to traditional methods (e.g., CCA + LDA), the Transformer-LSTM hybrid model achieved a relative improvement of over 15% in validation accuracy.

---

6. Challenges and Iterative Improvements

6.1 Challenges

- **Limited number of training samples** for some subjects.
- **Variability in EEG quality** across sessions.
- **FBCCA sensitivity** to electrode configurations or noise.
- **Computational cost** of testing multiple deep models.

6.2 Experiments Conducted (Even if not in final pipeline):

- **Feature Extraction Methods Tested:**
  - CCA only (baseline).
  - STFT & wavelet transform features.
  - FFT-based frequency-domain features.
- **Models Tested:**
  - LDA, SVM (with limited success).
  - DeepConvNet (high params, lower generalization).
  - Transformer-only (overfitting risk).
- **Preprocessing Variants:**
  - Using 3 vs. 5 vs. 7 subbands in FBCCA.
  - Trying different target frequencies (6–14 Hz).

- Normalization using RobustScaler instead of StandardScaler.

---

7. Conclusion

This SSVEP classification system combines **robust FBCCA-based feature extraction** with **powerful attention-based deep learning architectures**, achieving strong accuracy and reliable confidence scores. The modularity of the pipeline facilitated extensive experimentation, allowing us to identify the **TransformerLSTMHybrid** model as the most effective for this task.

Together with the MI pipeline, this contributed significantly to the overall mean accuracy in the AIC-3 competition.