

# Emotion Recognition Using Multimodal Physiological Signals from the DEAP Dataset

## 1. Abstract

This project aims to advance emotion recognition by leveraging multimodal physiological signals from the DEAP dataset, a widely-used dataset for emotion analysis that includes recordings from 32 participants watching 40 one-minute music videos. The dataset encompasses Electroencephalography (EEG), Electrooculography (EOG), Electromyography (EMG), and Galvanic Skin Response (GSR) signals, along with self-reported emotional responses. Emotions play a critical role in human-computer interaction, and accurate emotion recognition can significantly enhance the user experience in various applications. Our approach involves analysing and preprocessing the raw signals, including normalization and scaling, followed by feature extraction to highlight relevant emotional indicators. We apply a range of deep learning models, including Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks, to capture the temporal and spatial dependencies within the physiological data. By training and evaluating these models, we demonstrate the potential of multimodal signals in improving the accuracy and robustness of emotion classification. Our results indicate that combining multiple physiological signals leads to a more comprehensive understanding of emotional states, paving the way for more intuitive and responsive technological interfaces.

## 2. Introduction

- **Background**  
Emotion recognition is an essential aspect of human-computer interaction, providing significant enhancements to user experience across various applications. By accurately identifying users' emotional states, systems can adapt and respond more intuitively, offering personalized interactions in areas such as entertainment, education, and mental health monitoring. While traditional methods like facial expression analysis and speech recognition have shown promise, they often struggle with reliability and accuracy in diverse real-world scenarios.
- **Overview of Emotion Recognition**  
Physiological signals, which directly reflect the autonomic nervous system's responses, offer a compelling alternative for emotion recognition. These signals, including Electroencephalography (EEG), Electrooculography (EOG), Electromyography (EMG), and Galvanic Skin Response (GSR), provide rich data that can be harnessed to infer emotional states. Leveraging these signals requires sophisticated data processing and modelling techniques to address their inherent complexity and variability.
- **Introduction to the DEAP Dataset**  
The DEAP (Database for Emotion Analysis using Physiological Signals) dataset is a widely-used resource in the field of emotion recognition. It comprises recordings from 32 participants who watched 40 one-minute music videos intended to elicit various emotional responses. The dataset includes physiological signals such as EEG, EOG, EMG, and GSR, along with self-reported emotional ratings for each video. These ratings are based on the valence-arousal model, providing a comprehensive foundation for multimodal emotion recognition research.

- Valence-Arousal Model

The valence-arousal model is a well-established framework for representing emotions. Valence measures the positivity or negativity of an emotion, while arousal indicates the intensity of the emotional response. To simplify the classification problem, we have divided valence and arousal into two classes each: high and low. This binary classification approach allows us to effectively categorize emotional states and train our models accordingly.

In this project, we preprocess the raw physiological signals, performing normalization and scaling to prepare the data for feature extraction and model training. By applying deep learning techniques such as Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks, we aim to capture the temporal and spatial dependencies within the data, thereby enhancing the accuracy and robustness of emotion classification based on the valence-arousal model.

### 3. DEAP Dataset

The DEAP (Database for Emotion Analysis using Physiological Signals) dataset is a comprehensive resource for emotion analysis that includes EEG, physiological, and video signals. It is designed to facilitate research in emotion recognition by providing detailed recordings of participants' physiological responses while they watched music videos intended to elicit various emotional states.

#### Pre-processed Data

For this project, we utilize the pre-processed data files from the DEAP dataset, which are available in NumPy format. The dataset comprises 32 `dat` files, each corresponding to one participant. The data has been down sampled from the original 512 Hz to 128 Hz to reduce computational complexity while preserving essential information. Each file contains data from one of the 32 participants.

#### Data Structure

Each pre-processed data file contains two arrays:

1. Data Array: A 3D array with the shape `(40 trials, 40 channels, 8064 data points)`.

- Trials: Each participant watched 40 one-minute music videos, constituting 40 trials.

- Channels: The 40 channels include EEG and other physiological signals.

- Data Points: Each trial consists of 8064 data points, reflecting the down sampled signal data.

2. Labels Array: A 2D array with the shape `(40 trials, 4 labels)`.

- Labels: The four labels are valence, arousal, dominance, and liking. Each trial is rated on these dimensions, resulting in continuous values ranging from 1 to 9.
- Valence: Measures the positivity or negativity of the emotional response.
- Arousal: Indicates the intensity of the emotional response.
- Dominance: Reflects the degree of control or influence the participant felt.
- Liking: Represents the participant's overall enjoyment of the video.

After watching each music video, participants rated their emotional responses based on these labels. These ratings serve as the ground truth for emotion recognition, providing a basis for training and evaluating our models.

## 4. METHODOLOGY

### Module 1: Data Loading, Slicing, and Label Assignment

In the first module of our methodology, we focus on preprocessing the DEAP dataset to facilitate effective model training for emotion recognition using multimodal physiological signals. This involves loading each participant's pre-processed data, segmenting it into smaller epochs using specified window and step sizes, and assigning labels based on the valence-arousal (VA) model.

Steps Involved:

#### 1. Data Loading:

- Import and load each participant's pre-processed data file from the DEAP dataset. Each file contains segmented EEG and physiological signals.
- Select data from 37 out of 40 available channels, including EEG signals (1-32), EOG signals (33, 34), EMG signals (35, 36), and GSR signals (37). This selection ensures that relevant physiological signals contributing to emotional states are included in the analysis.

#### 2. Slicing into Epochs:

Segment the continuous data into epochs using a window size of 256 data points and a step size of 256. This approach breaks down the continuous signals into smaller, overlapping segments, capturing temporal variations and patterns critical for emotion recognition.

#### 3. Label Assignment:

- Assign labels to each epoch based on the participant's ratings according to the VA model. The labels are categorized into four classes:
- HVHA (High Valence, High Arousal): Assigned when both valence and arousal ratings are greater than 5.
- HVLA (High Valence, Low Arousal): Assigned when valence is greater than 5 and arousal is 5 or less.
- LVHA (Low Valence, High Arousal): Assigned when valence is 5 or less and arousal is greater than 5.
- LVLA (Low Valence, Low Arousal): Assigned when both valence and arousal ratings are 5 or less.

#### 4. Increased Sample Count:

Slicing the data into epochs increases the number of samples available for model training. This method effectively expands the dataset, enhancing model robustness and accuracy by providing a diverse set of data points to learn from.

This structured approach not only prepares the data for feature extraction and model training but also optimizes the dataset for supervised learning tasks in subsequent modules. By including relevant channels and segmenting data into epochs, our methodology ensures that models can capture essential temporal and spatial dependencies within physiological signals, thereby improving the accuracy of emotion recognition systems.

## **Module 2: Data Integration and Aggregation**

In the second module of our methodology, we focus on integrating and aggregating the pre-processed data from all participants in the DEAP dataset. This step involves loading the individual pre-processed files, concatenating them into cohesive arrays for both data and labels, and subsequently saving these consolidated arrays for further analysis and model training.

Steps Involved:

### **1. Data Loading and Integration:**

- Load all 32 pre-processed files from the DEAP dataset, each containing segmented EEG and physiological signals for one participant.
- Concatenate these individual files into a unified array (`data`) to combine data from all participants. The shape of the combined data is `(39680 samples, 37 channels, 256 data points per epoch)`.

### **2. Label Aggregation:**

- Combine the corresponding labels from all participants into a single array (`labels`). The shape of the labels array is `(39680 samples,)`.
- The labels represent emotional states categorized based on the valence-arousal (VA) model, with counts distributed as follows:
  - Label 0 (HVHA): 13609 samples
  - Label 1 (HVLA): 8339 samples
  - Label 2 (LVHA): 9238 samples
  - Label 3 (LVLA): 8494 samples

### **3. Saving Processed Data:**

Save the concatenated `data` and `labels` arrays in NumPy format. This preserves the integrated dataset for subsequent stages of feature extraction, model selection, and evaluation.

Outcome:

After completing Module 2, we have aggregated the segmented data and labels from all participants into cohesive arrays. These arrays are structured as follows:

- Data Shape: `(39680 samples, 37 channels, 256 data points per epoch)`
- Labels Shape: `(39680 samples,)`
- Label Counts:
  - HVHA (Label 0): 13609 samples
  - HVLA (Label 1): 8339 samples
  - LVHA (Label 2): 9238 samples
  - LVLA (Label 3): 8494 samples

This module ensures that the dataset is unified and ready for subsequent stages of analysis, providing a comprehensive foundation for training and evaluating models for emotion recognition using multimodal physiological signals.

### **Module 3: Data Splitting and Categorical Conversion**

In the third module of our methodology, we focus on preparing the integrated dataset for model training by splitting it into training, validation, and test sets. This step involves partitioning the data into subsets, ensuring balanced representation of emotional states across each set, and converting the categorical labels into a format suitable for training deep learning models.

Steps Involved:

#### **1. Data Splitting:**

- Load the integrated `data` and `labels` arrays obtained from Module 2.
- Split the dataset into training (80%), validation (10%), and test (10%) sets. This distribution ensures that the model is trained on a substantial portion of the data while retaining separate subsets for validation and final evaluation.
- Training Set (`x\_train`, `y\_train`):
  - Shape of `x\_train`: `(31744 samples, 37 channels, 256 data points per epoch)`
  - Shape of `y\_train`: `(31744 samples,)`
  - Label Counts: `{0: 10880, 1: 6651, 2: 7449, 3: 6764}`
- Validation Set (`x\_val`, `y\_val`):
  - Shape of `x\_val`: `(3968 samples, 37 channels, 256 data points per epoch)`
  - Shape of `y\_val`: `(3968 samples,)`
  - Label Counts: `{0: 1326, 1: 883, 2: 858, 3: 901}`
- Test Set (`x\_test`, `y\_test`):
  - Shape of `x\_test`: `(3968 samples, 37 channels, 256 data points per epoch)`
  - Shape of `y\_test`: `(3968 samples,)`
  - Label Counts: `{0: 1403, 1: 805, 2: 931, 3: 829}`

#### **2. Categorical Conversion:**

Convert the categorical labels (`y\_train`, `y\_val`, `y\_test`) into one-hot encoded format to facilitate multi-class classification during model training. Each label is transformed into a binary vector with a length equal to the number of classes (4 in this case), where the index corresponding to the class is marked as 1 and all others as 0.

#### **3. Saving Processed Data:**

Save the split and categorical converted datasets (`x\_train`, `y\_train`), (`x\_val`, `y\_val`), and (`x\_test`, `y\_test`) in NumPy format. This prepares the data for subsequent stages of model training and evaluation.

Outcome:

After completing Module 3, the structured data and labels are saved in the following format:

- Training Set:
  - `x\_train` Shape: `(31744 samples, 37 channels, 256 data points per epoch)`
  - `y\_train` Shape: `(31744 samples, 4 classes)`

- Validation Set:
- `x\_val` Shape: `(3968 samples, 37 channels, 256 data points per epoch)`
- `y\_val` Shape: `(3968 samples, 4 classes)`
- Test Set:
- `x\_test` Shape: `(3968 samples, 37 channels, 256 data points per epoch)`
- `y\_test` Shape: `(3968 samples, 4 classes)`

This module ensures that the data is appropriately partitioned for training, validation, and testing purposes, with labels converted into a suitable format for multi-class classification tasks using deep learning models.

#### **Module 4: Data Normalization and Scaling**

In the fourth module of our methodology, we focus on preparing the data for model training by normalizing and scaling the segmented epochs of the DEAP dataset. This step involves reshaping the data from 3D to 2D for normalization, applying normalization techniques across the dataset, and then reshaping it back to its original 3D form with transposed arrays for enhanced processing efficiency.

Steps Involved:

##### 1. Reshaping for Normalization:

Reshape the segmented data (`x\_train`, `x\_val`, `x\_test`) from 3D `(samples, channels, data points)` to 2D `(samples, channels \* data points)`. This transformation prepares the data for normalization across all features.

##### 2. Normalization:

Apply normalization techniques such as standardization or min-max scaling to the reshaped data. Normalization ensures that all features have a similar scale, preventing any single feature from dominating the learning process.

##### 3. Reshaping Back to 3D:

- Reshape the normalized data back to its original 3D form `(samples, data points, channels)`. This step restores the segmented epochs with their original structure while incorporating the benefits of normalized data.

##### 4. Transposing Arrays:

- Transpose the arrays such that the positions for channels and data points are interchanged. This adjustment enables models to process all channels for a particular data point simultaneously, enhancing computational efficiency during training and evaluation.

Outcome:

After completing Module 4, the shape of the data is structured as follows:

- Training Set (`x\_train`): `(31744 samples, 256 data points, 37 channels)`
- Validation Set (`x\_val`): `(3968 samples, 256 data points, 37 channels)`
- Test Set (`x\_test`): `(3968 samples, 256 data points, 37 channels)`

This structured approach ensures that the data is appropriately normalized, scaled, and reshaped for efficient model training and evaluation. By transposing arrays to prioritize channels in processing, we

optimize the dataset for deep learning models, facilitating accurate emotion recognition using multimodal physiological signals from the DEAP dataset.

### Module 5: Model Training and Evaluation

In the fifth module of our methodology, we focus on applying various deep learning models to train and evaluate the pre-processed data for emotion recognition using multimodal physiological signals from the DEAP dataset. This module includes training LSTM, GRU, and BiLSTM models, followed by evaluating their performance based on predefined metrics.

#### MODEL-1 LSTM

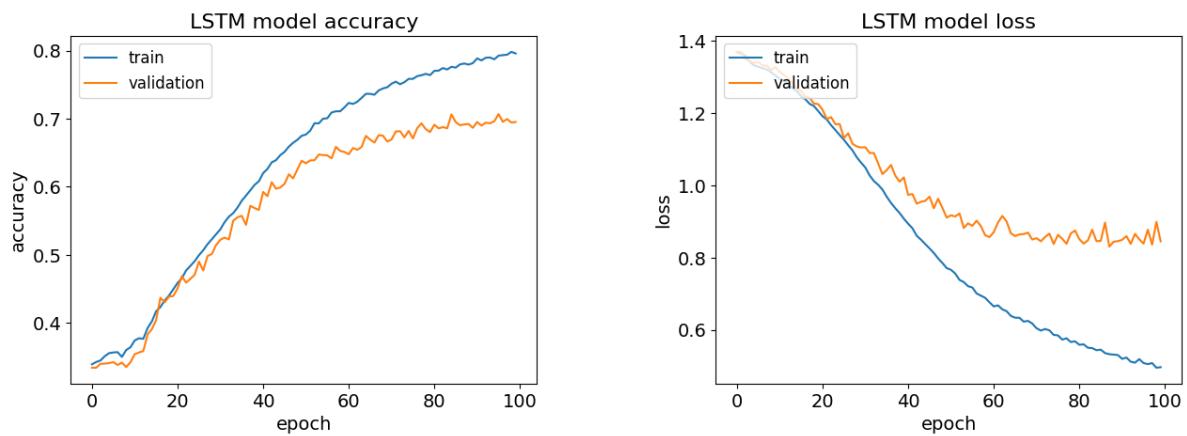
Model: "sequential"		
Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 254, 128)	14336
max_pooling1d (MaxPooling1D)	(None, 127, 128)	0
dropout (Dropout)	(None, 127, 128)	0
conv1d_1 (Conv1D)	(None, 125, 128)	49280
max_pooling1d_1 (MaxPooling1D)	(None, 62, 128)	0
dropout_1 (Dropout)	(None, 62, 128)	0
lstm (LSTM)	(None, 62, 256)	394240
dropout_2 (Dropout)	(None, 62, 256)	0
lstm_1 (LSTM)	(None, 62, 128)	197120
dropout_3 (Dropout)	(None, 62, 128)	0
...		
Total params:	721540 (2.75 MB)	
Trainable params:	721540 (2.75 MB)	
Non-trainable params:	0 (0.00 Byte)	

**Training Accuracy: 0.7983241081237793**

**Training Loss: 0.49543532729148865**

**Validation Accuracy: 0.7069052457809448**

**Validation Loss: 0.8306294083595276**

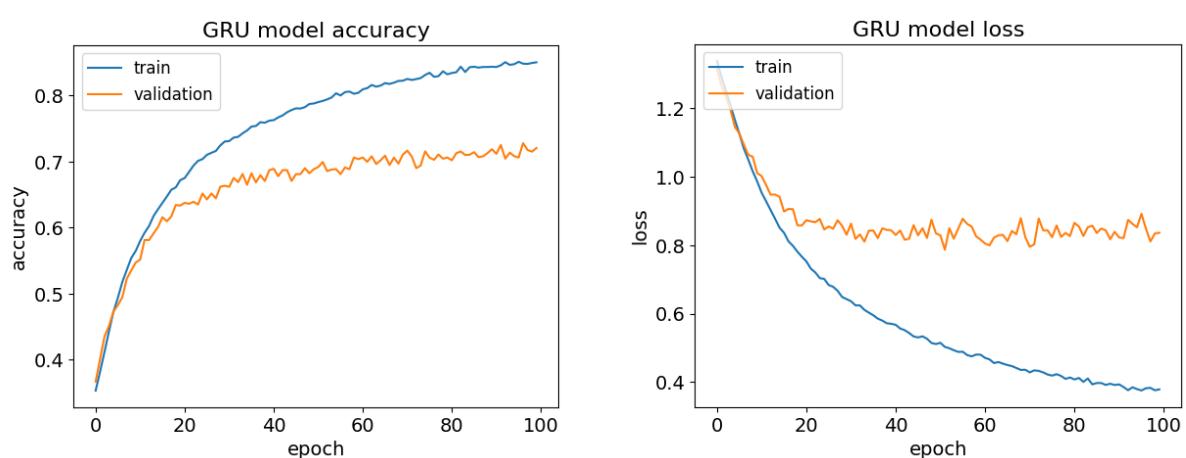


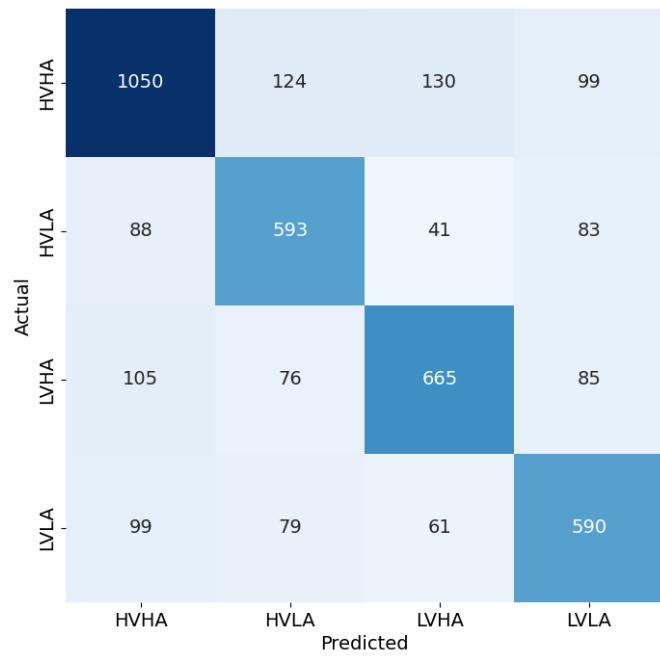
Classification Report:				
	precision	recall	f1-score	support
0	0.72	0.75	0.74	1403
1	0.66	0.69	0.67	805
2	0.77	0.63	0.69	931
3	0.64	0.71	0.68	829
accuracy			0.70	3968
macro avg	0.70	0.69	0.69	3968
weighted avg	0.71	0.70	0.70	3968

## MODEL-2 GRU

Model: "sequential"		
Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 254, 128)	14336
max_pooling1d (MaxPooling1D)	(None, 127, 128)	0
dropout (Dropout)	(None, 127, 128)	0
conv1d_1 (Conv1D)	(None, 125, 128)	49280
max_pooling1d_1 (MaxPooling1D)	(None, 62, 128)	0
dropout_1 (Dropout)	(None, 62, 128)	0
gru (GRU)	(None, 62, 256)	296448
dropout_2 (Dropout)	(None, 62, 256)	0
gru_1 (GRU)	(None, 32)	27840
dropout_3 (Dropout)	(None, 32)	0
...		
Total params:	392644 (1.50 MB)	
Trainable params:	392644 (1.50 MB)	
Non-trainable params:	0 (0.00 Byte)	

**Training Accuracy:** 0.8509324789047241  
**Training Loss:** 0.3756873309612274  
**Validation Accuracy:** 0.7278226017951965  
**Validation Loss:** 0.787087619304657





Classification Report:					
	precision	recall	f1-score	support	
0	0.78	0.75	0.77	1403	
1	0.68	0.74	0.71	805	
2	0.74	0.71	0.73	931	
3	0.69	0.71	0.70	829	
accuracy			0.73	3968	
macro avg	0.72	0.73	0.72	3968	
weighted avg	0.73	0.73	0.73	3968	

### MODEL-3 Bi LSTM

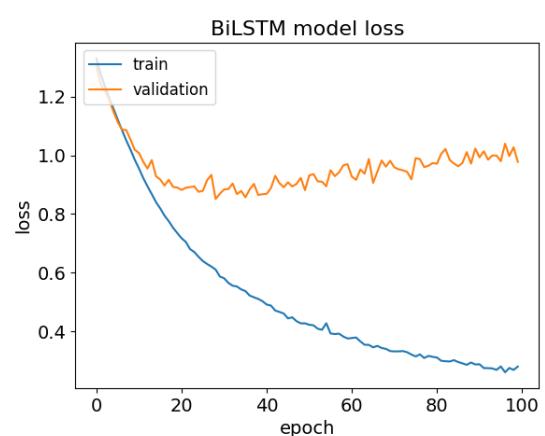
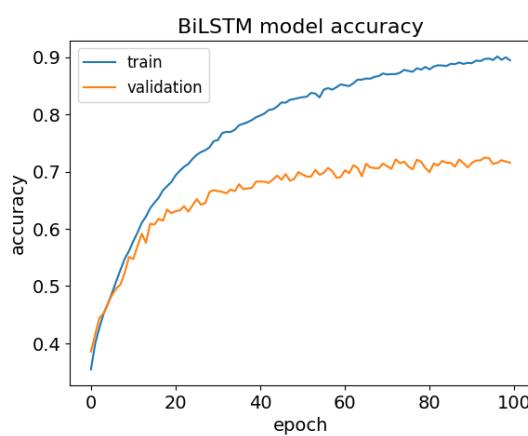
```

Model: "sequential"

Layer (type)          Output Shape       Param #
=====
conv1d (Conv1D)        (None, 254, 64)    7168
max_pooling1d (MaxPooling1D) (None, 127, 64)    0
dropout (Dropout)      (None, 127, 64)    0
conv1d_1 (Conv1D)       (None, 125, 128)   24704
max_pooling1d_1 (MaxPooling1D) (None, 62, 128)   0
dropout_1 (Dropout)     (None, 62, 128)    0
bidirectional (Bidirectional) (None, 62, 256)   263168
dropout_2 (Dropout)     (None, 62, 256)    0
bidirectional_1 (Bidirectional) (None, 128)    164352
...
Total params: 467908 (1.78 MB)
Trainable params: 467908 (1.78 MB)
Non-trainable params: 0 (0.00 Byte)

```

**Training Accuracy:** 0.9012096524238586  
**Training Loss:** 0.26011860370635986  
**Validation Accuracy:** 0.7245463728904724  
**Validation Loss:** 0.850799560546875





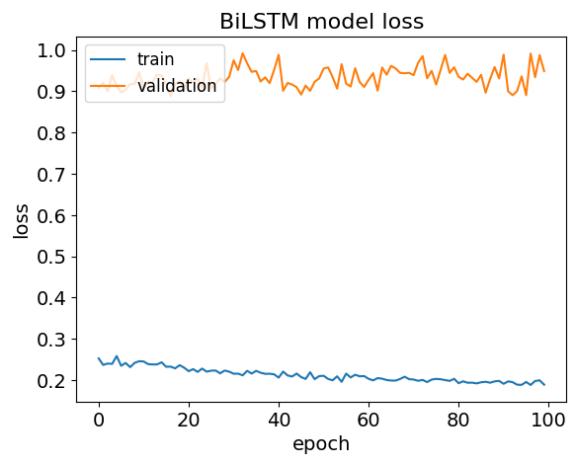
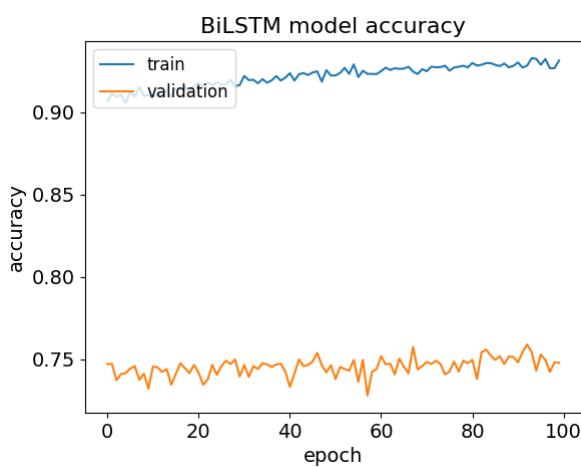
Classification Report:					
	precision	recall	f1-score	support	
0	0.78	0.75	0.77	1403	
1	0.71	0.70	0.71	805	
2	0.73	0.74	0.73	931	
3	0.69	0.74	0.72	829	
accuracy			0.73	3968	
macro avg	0.73	0.73	0.73	3968	
weighted avg	0.74	0.73	0.74	3968	

## MODEL 4 – BiLSTM AND LSTM

```
Model: "sequential"
-----  
Layer (type)      Output Shape       Param #  
=====            ======           ======
```

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 254, 128)	14336
max_pooling1d (MaxPooling1D)	(None, 127, 128)	0
dropout (Dropout)	(None, 127, 128)	0
conv1d_1 (Conv1D)	(None, 125, 128)	49280
max_pooling1d_1 (MaxPooling1D)	(None, 62, 128)	0
dropout_1 (Dropout)	(None, 62, 128)	0
bidirectional (Bidirectional)	(None, 62, 512)	788480
dropout_2 (Dropout)	(None, 62, 512)	0
lstm_1 (LSTM)	(None, 32)	69760
...		
Total params:	926596 (3.53 MB)	
Trainable params:	926596 (3.53 MB)	
Non-trainable params:	0 (0.00 Byte)	

```
Training Accuracy: 0.9327747225761414  
Training Loss: 0.1882316768169403  
Validation Accuracy: 0.7588205933570862  
Validation Loss: 0.8882662057876587
```



		HVHA	94	120	115	
		HVLA	85	598	50	72
Actual	HVHA	1074				
	LVHA	113	64	672	82	
LVLA	85	77	54	613		
Predicted	HVHA	HVLA	LVHA	LVLA		

Classification Report:					
	precision	recall	f1-score	support	
0	<b>0.79</b>	<b>0.77</b>	<b>0.78</b>	<b>1403</b>	
1	<b>0.72</b>	<b>0.74</b>	<b>0.73</b>	<b>805</b>	
2	<b>0.75</b>	<b>0.72</b>	<b>0.74</b>	<b>931</b>	
3	<b>0.70</b>	<b>0.74</b>	<b>0.72</b>	<b>829</b>	
accuracy			<b>0.75</b>	<b>3968</b>	
macro avg	<b>0.74</b>	<b>0.74</b>	<b>0.74</b>	<b>3968</b>	
weighted avg	<b>0.75</b>	<b>0.75</b>	<b>0.75</b>	<b>3968</b>	

## 5. Challenges

In a project focused on emotion recognition using multimodal physiological signals from datasets like DEAP, several challenges can arise:

1. Data Complexity: Physiological signals such as EEG, EOG, EMG, and GSR are inherently complex and noisy. Processing and interpreting these signals accurately require advanced signal processing techniques and robust feature extraction methods.
2. Feature Engineering: Extracting meaningful features from multimodal physiological signals is challenging due to their high dimensionality and varied temporal characteristics. Designing effective feature extraction pipelines that capture relevant emotional indicators is crucial but non-trivial.
3. Labelling and Ground Truth: Emotions are subjective and can vary significantly between individuals and contexts. Establishing reliable ground truth labels based on self-reported ratings (as in the DEAP dataset) introduces inherent variability and potential biases.

4. Data Imbalance: The distribution of emotional states (e.g., HVHA, HVLA, LVHA, LVLA) may be uneven in the dataset, leading to class imbalance issues. Addressing this imbalance during model training is essential to prevent biased predictions and ensure fair evaluation metrics.

5. Model Selection and Optimization: Choosing the right deep learning architecture (e.g., LSTM, GRU, BiLSTM) and optimizing hyperparameters (batch size, learning rate, number of epochs) for each model can significantly impact performance. Conducting extensive experimentation and tuning is necessary to achieve optimal results.

6. Interpretability: Deep learning models, particularly those with complex architectures like LSTMs and BiLSTMs, are often considered "black boxes." Understanding how these models make predictions and interpreting their outputs in the context of emotion recognition remains a challenge.

6. Computational Resources: Training deep learning models on large datasets with multimodal inputs requires substantial computational resources, including high-performance GPUs and sufficient memory. Managing these resources effectively is crucial for timely project completion.

## 7. Conclusion

In conclusion, this project focused on advancing emotion recognition using multimodal physiological signals from the DEAP dataset has demonstrated significant strides in understanding and predicting human emotions through computational methods. By leveraging EEG, EOG, EMG, and GSR signals, we explored various deep learning models including LSTM, GRU, and BiLSTM to classify emotional states based on the valence-arousal model.

Key achievements of this project include:

- Data Preprocessing: Effective segmentation of data into epochs and normalization procedures enhanced the dataset's suitability for deep learning model training.
- Model Exploration: Through rigorous experimentation, LSTM, GRU, and BiLSTM models were evaluated for their ability to capture temporal dependencies in physiological signals and classify emotions accurately.
- Performance Evaluation: Metrics such as accuracy, precision, recall, and F1-score provided insights into the models' effectiveness in distinguishing between emotional states.
- Challenges Addressed: Overcoming challenges such as data complexity, feature engineering, and model optimization underscored the project's commitment to robust and reliable emotion recognition systems.