Agenda

- Abstract
- Problem Statement
- Literature Survey
- Result of Previous Work
- Gaps Identified
- Proposed Methodology
- Dataset Analysis
- Model Architecture
- Cross-validation
- Proposed Method - 1
- Proposed Method - 2
- Result Analysis
- Requirements
- Conclusion
- Future Enhancement
- Reference

Abstract

- **Knowledge distillation (KD)** is a widely-used machine learning technique designed to transfer knowledge from a large 'teacher' network to a smaller 'student' network, enhancing the student's performance by training it to mimic the teacher.
- Similarly, **transfer learning (TL)** is effective in accelerating the training of neural networks on limited datasets by leveraging representations learned from related tasks.
- A deep learning architecture, referred to as **TL+KD**, **integrates knowledge distillation with transfer learning** and allows for a quantitative and qualitative comparison with standalone TL and KD in **fear level classification**.
- By incorporating guidance from a larger teacher network during fine-tuning, the student network demonstrates improved validation performance, such as higher accuracy.
- Furthermore, this approach enhances model performance, as reflected in multiple evaluation metrics beyond just accuracy.

Problem Statement

- Existing emotion recognition models, particularly for fear level detection using physiological signals (EEG, ECG, GSR, etc.), often face challenges with accuracy and generalization across datasets and populations.
- To address these limitations, this project aims to develop a knowledge distillation and transfer learning-based solution, enhancing feature extraction and improving model performance for reliable and scalable fear detection.

Objectives

- To enhance the accuracy and generalizability of fear detection using **computational intelligence.**
- To implement a comprehensive strategy that includes exploring **effects of Knowledge Distillation on Transfer Learning, integrating multimodal data and preprocessing techniques**, and ensuring model robustness across diverse datasets.
- This approach is designed to **address existing gaps** in the current solutions and push the boundaries of fear detection technology.

Literature Survey

S.NO	PAPER	DATASET	MODELS	INPUTS	INFERENCE	ACCURACY
1	Title "Emotion Classification Based on Transformer and CNN for EEG Spatial-Temporal Feature Learning" Author - Xiuzhen Yao, Tianwen Li, Peng Ding, Year - 2024	DEAP , SEED Dataset	EEG ST-TCNN	EEG Signals	The study employed a Transformer CNN model to accurately estimate Emotion for EEG spatial–temporal (EEG ST) feature learning to automatic emotion classification..	96.34% for SEED. And 96.95%, and 96.34% for the arousal, and valence for DEAP
2	Title: "EmotionKD:A Cross-Modal Knowledge Distillation Framework for Emotion Recognition Based on Physiological Signals", Proceedings of the 31st ACM International Conference on Multimedia, Year 2023	DEAP Dataset	EmotionNet-Teacher (multimodal) and EmotionNet-Student (unimodal)	EEG and GSR signals	Knowledge distillation transfers fused multi-modal EEG and GSR features to a unimodal GSR model for better emotion recognition	69.18%
3	Title: " Emotion and Personality Recognition Using Commercial Sensors (IEEE Transactions on Affective Computing),Year 2018.	ASCERTAIN Dataset	SVM KNN RF	EEG, ECG and GSR Signals	Emotion recognition (valence/arousal classification) is more accurate when EEG signals are used compared to other modalities.	81.18%

Literature Survey

4	Title"Multi-Domain Feature Fusion for Emotion Classification Using DEAP Dataset" Authors Digital Object Identifier M. Alarcao, and Manuel J. Fonseca, Ieee Access, Year 2021	DEAP Dataset	SVM	EEG Signals	Classification model achieved the highest (with Hjorth and classification on DEAP dataset to the best of our knowledge.	94%
5	Title "Human Emotion Recognition Using Machine Learning Techniques Based on Physiological Signals Published in Biomedical Signal Processing and Control, Part A, Year 2025	ASCERTAIN Dataset	KNN RF	GSR EEG ECG Signals	The Random Forest classifier demonstrated superior performance for EEG and GSR signals in the ASCERTAIN datasets.	76%
6	Title: "Stress Detection with Machine Learning and Deep Learning Using Multimodal Data" Author Sorasa M. Alarcao, and Manuel J. Fonseca, (ICIRCA). IEEE, Year 2020	DEAP Dataset	RNN LSTM GRU	EEG Signals	The most used feature selection algorithms are (PCA), (SFS), Model of Affect—valence, arousal, dominance, liking etc.	95%

Literature Survey

7	Title"An Investigation of Various Automatic Fear Detection Authors: Oana Bălan, Gabriela Moise, Alin Moldoveanu, Sensors, Year 2020	MAHNOB Dataset	SVM, Decision Tree, Random Forest	EEG, HR,GSR Signals	The Method defines a complex emotion such as fear as a combination of the data into low or high valence, arousal and dominance.	89%
8	Title"An Investigation of Various Machine and Deep Learning in Fear Authors - Oana Bălan , Gabriela Moise , Alin Moldoveanu, Sensors, Year 2020	DEAP Dataset	SVM RF LDA	EEG Signals	The study employed a VR system that captures levels and predict appropriate exposure scenarios. cross-validation and test accuracies.	79%
9	Title: "Comparative Analysis of machine learning using EEG Author Vikrant Doma and Matin Pirouz Journal of Big Data, Year 2020"	DEAP Dataset	SVM KNN	BEG and Peripheral Physiological signals	The study utilized machine learning models like SVM, KNN, Logistic Regression, and preprocessing for improved emotion recognition accuracy.	75%

Result of Previous Work

Both Teacher and Student models are trained for 100 epochs, and student model has reduced parameters (para.) and inputs. Threshold - accuracy > 60%.

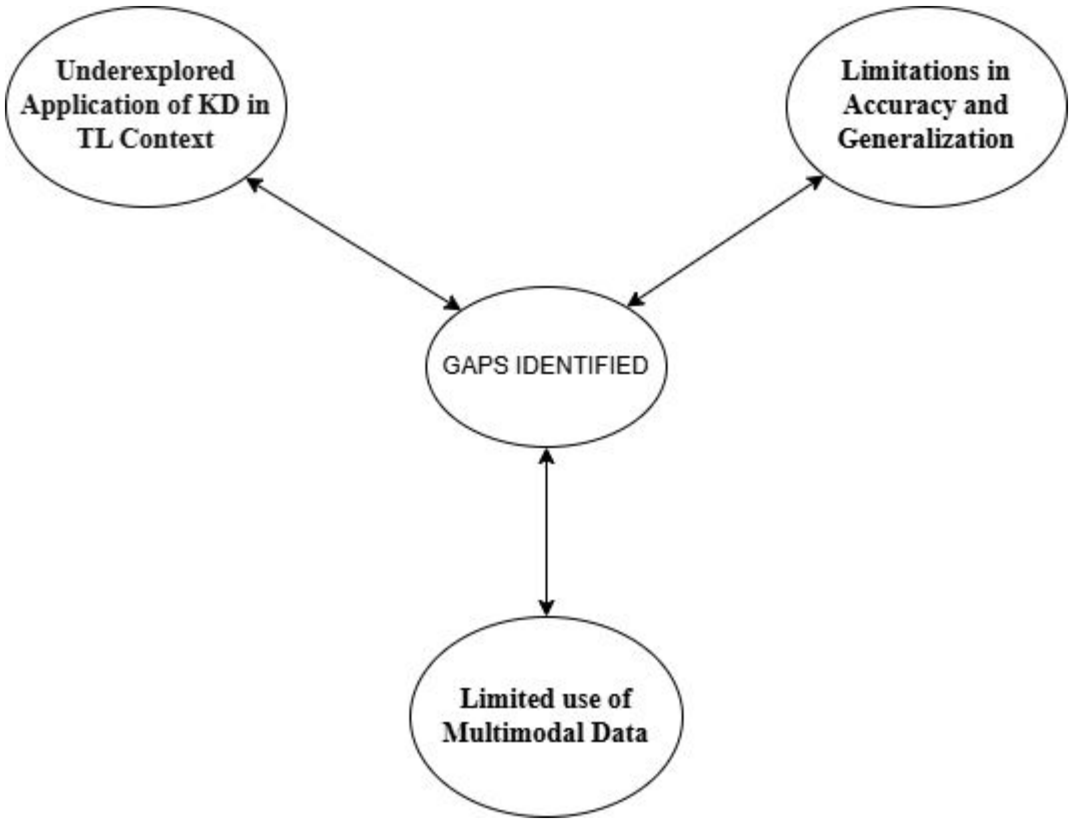
Model Name	With Knowledge Distillation							
	Teacher Model			Student Model				
	Channels	Para.	Testing Accuracy (%)	Stud. Model	Channels	Para.	Without KD Test Acc. (%)	With KD Test Accuracy (%)
ANN	40	4,836	57.51	ANN	8	420	45.38	49.71
DNN 150	40	52,054	69.94	DNN 150	8	24,604	67.73	58.09
DNN 300	40	374,704	69.36	DNN 300	8	274,804	63.29	**61.27**
LSTM	40	19,572	51.16	LSTM	8	4,948	48	48
CNN-LSTM Hybrid	40	148,996	91.33	DNN 300	8	184,504	63.29	**64.45**
				LSTM	8	35,684	48	51.73
CNN	40	875,342	92.03	CNN	8	150,158	79.19	**76.01**

Result of Previous Work

Knowledge Distillation was applied on Student Model with 40 channels(Ch.) as inputs.

Model Name	With Knowledge Distillation							
	Teacher Model			Student Model				
	Ch.	Para.	Testing Accuracy (%)	Model	Ch.	Para.	Without KD Test Acc. (%)	With KD Test Acc. (%)
DNN 300	40	1,124,114	75.43	DNN 300	8	824,414	73.41	70.23
					40	853,214	75.43	73.12
CNN-LSTM Hybrid	40	446,990	91.90	DNN 300	8	184,504	53.17	54.62
					40	382,314	53.75	55.49
CNN	40	875,342	92.03	CNN	8	88,910	76.30	77.16
					40	814,094	89.1	93.03

Gaps Identified



Gaps Identified

- **Underexplored Application of KD in TL Context**
  - The combined application of KD and TL for emotion recognition, especially in fear level detection, is **underexplored**
- **Limitations in Accuracy and Generalization**
  - Many models fail to **adapt to the variability** in physiological signals arising from individual differences, noise, and inconsistencies in the dataset.
- **Limited Use of Multimodal Data**
  - Insufficient integration of multimodal data to exploit **complementary features** that enhance model performance and robustness.

Proposed Methodology

- **Knowledge Distillation: Teacher-Student Approach**
  - Knowledge distillation allows the **transfer** of knowledge from a large, complex model (**teacher**) trained on multimodal data to a simpler, smaller model (**student**).
  - **Fewer input** channels in **student models** reduce computational load, enabling faster, efficient real-time applications with cost effectiveness

- **Transfer Learning: Utilising pre-trained models**
  - Transfer learning enables the **reuse** of knowledge from **pre-trained** models on emotion recognition, **reducing** the need for large f**ear-specific datasets** and accelerating training for fear level detection.
  - By leveraging features learned from broader emotional datasets, transfer learning enhances the model's ability to detect subtle **fear patterns**, improving **accuracy**

Proposed Methodology

- **Knowledge Distillation on Transfer Learning: Novel Approach**
  - Knowledge distillation **simplifies complex models** by transferring knowledge to smaller models, improving efficiency in fear detection.
  - Combining distillation with transfer learning enhances the model's ability to **generalize across datasets**, crucial for fear level detection.
  - Knowledge distillation helps the student model **learn important features** from the teacher model, boosting fear-level classification accuracy.
  - Transfer learning accelerates the **adaptation of models** to new fear detection tasks while distillation **refines performance** using fewer resources.

Proposed Methodology

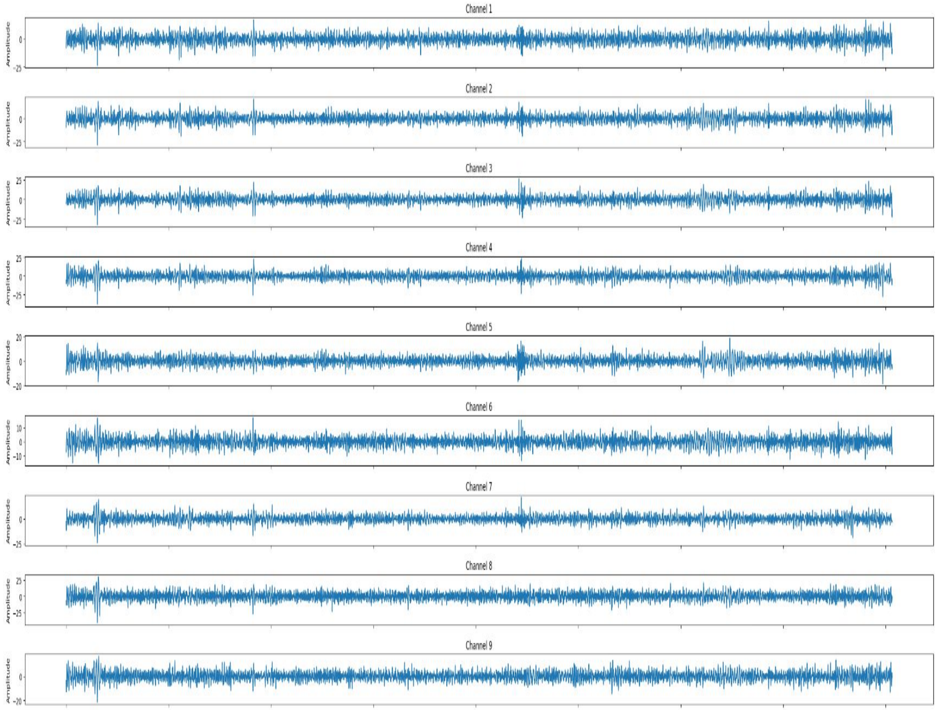
- **Model Variety: Exploring Ensemble and Hybrid Models**
  - To leverage the strengths of **multiple models** to improve prediction accuracy.
  - Compare the performance of these models using **cross-validation** to ensure robust evaluation.

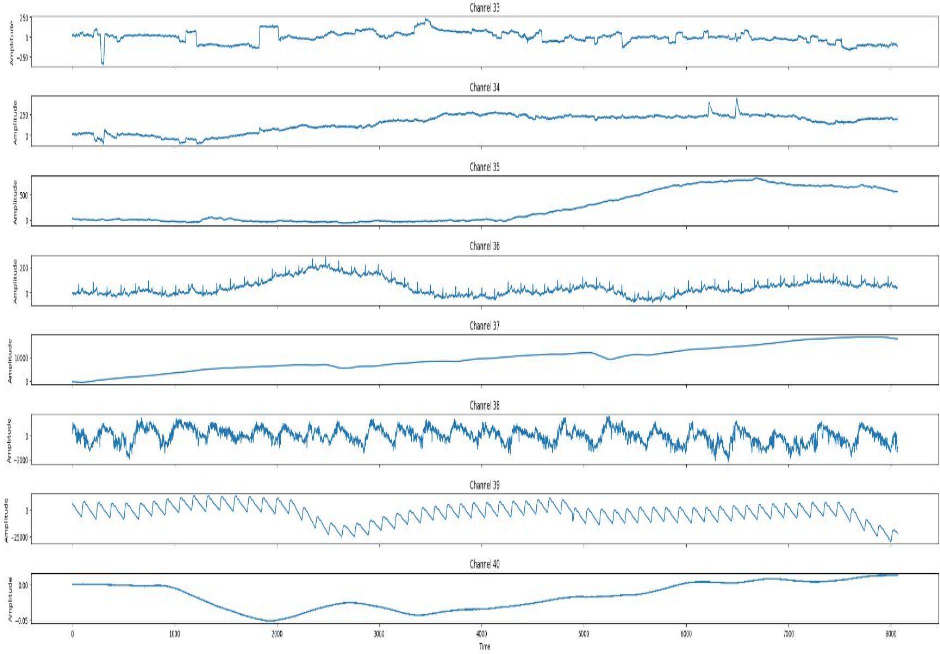
- **Multi-Modal Integration: Combining Additional Modalities**
  - To improve model performance by integrating **data from multiple sources**, such as EEG, ECG, and other physiological signals(hEOG,vEOG,zEMG,GSR, Respiration, PPG, Temperature).
  - Implement **multi-input models** that can process and integrate these different data sources.
  - Evaluate the impact of multi-modal integration on model performance.

DEAP Dataset Analysis

- **Name**: DEAP Dataset
- **Purpose**: Emotion analysis and classification using physiological signals.
- **Dataset Type**: Multimodal physiological dataset (EEG, EMG, GSR, ECG, etc.)
- **Size**: 1.9 GB
- **Participants**: 32 subjects (16 males, 16 females).
- **Recording Duration**: Each participant's EEG signals were recorded for 63 seconds per video clip (60 seconds of stimuli and 3 seconds pre-trial baseline).
- **Sampling Rate**: 512 Hz (downsampled to 128 Hz for preprocessed data).
- **Total Data Points**: 32 participants × 40 video trials per participant = 1,280 trials.

DEAP Dataset Analysis





Fear Data Extraction Process - DEAP Dataset:

Label	Valence	Arousal	Dominance
No Fear (0)	[7:9]	[1:3)	[7:9]
Low Fear (1)	[5:7)	[3:5)	[5:7)
Medium Fear (2)	[3:5)	[5:7)	[3:5)
High Fear (3)	[1:3)	[7:9]	[1:3)

Cont...

- Import the preprocessed DEAP dataset from .mat files, which contain EEG signals and their corresponding valence, arousal, and dominance ratings for each trial.
- Filter the trials based on the given valence, arousal, and dominance ranges for different fear levels.
- Select EEG signals corresponding to the identified trials for each fear level and store them separately for feature extraction.
- Assign the appropriate fear level labels (0-3) to the extracted data and save it for training deep learning models for fear level classification.

Frequency Bands used in Extraction Process:

- EEG signals are decomposed into different frequency bands using band pass filtering, which are analyzed to extract meaningful features. Here five (5) frequency bands were used.

1. Theta Band (4-8 Hz)

- **Associated with**: Relaxation, creativity, drowsiness, early sleep stages.
- **Importance in Emotion Recognition**: Higher theta power is often linked to deep relaxation and meditation. It plays a role in emotional processing and memory formation.

2. Alpha Band (8-12 Hz)

- **Associated with**: Calmness, alert relaxation, wakeful rest.
- **Importance in Emotion Recognition**: Increased alpha activity is observed during relaxed, non-aroused states. It helps differentiate between neutral and emotionally charged states.

Frequency Bands used in Extraction Process:

3. Low Beta Band (12-16 Hz)

- **Associated with**: Focus, alertness, cognitive processing.
- **Importance in Emotion Recognition**: A moderate increase in beta activity is often linked to engagement, attention, and problem-solving tasks.

4. High Beta Band (16-25 Hz)

- **Associated with**: Stress, anxiety, high cognitive load.
- **Importance in Emotion Recognition**: High beta activity is often seen during stress or heightened emotional states.

5. Gamma Band (25-45 Hz)

- **Associated with**: Conscious perception, sensory processing, high-level cognitive functions.
- **Importance in Emotion Recognition**: Gamma waves are crucial in processing emotions and higher cognitive tasks.

Processing of PPS Signals:

PPS Channels Used

- Selected PPS channels: **hEOG, vEOG, zEMG(Electromyography), tEMG, GSR, Respiration, Plethysmograph, Temperature**

Feature Extraction from PPS Signals

- Iterated over each PPS channel.
- Extracted statistical features:
  - **Mean (pps_mean)** → np.mean(X)
  - **Variance (pps_var)** → np.var(X)
- Combined size - (32 EEG x 5 bands) + (8 PPS x 2 features) = **176 features**
- Dependent data, y = n x 10 (valence from 0 to 9)

ASCERTAIN Dataset:

- **ASCERTAIN (Affective Computing and Emotion Recognition) Dataset** is a widely used dataset in affective computing and emotion recognition research.
- It is designed to study human emotions using multimodal signals, including physiological and behavioral data.
- The dataset is collected to assess emotional states in response to multimedia stimuli, making it useful for research in emotion classification, affective computing, and brain-computer interface (BCI) applications.
- To capture these emotions effectively, multimodal data was recorded, including physiological signals such as EEG (8-channel brain activity at 256 Hz), ECG (heart activity at 256 Hz), and GSR (skin conductance at 8 Hz).

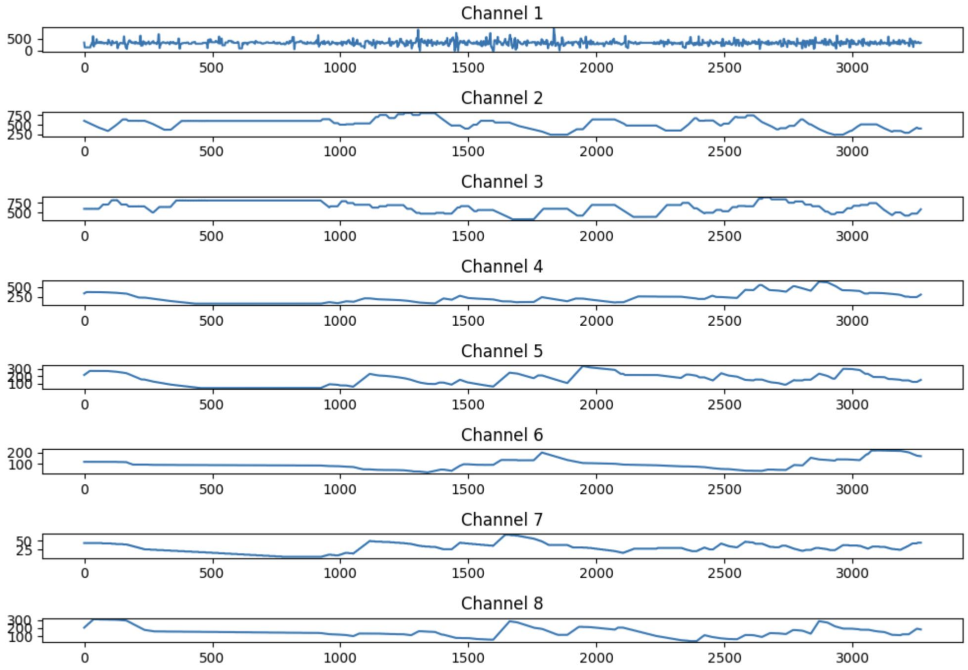
ASCERTAIN Dataset

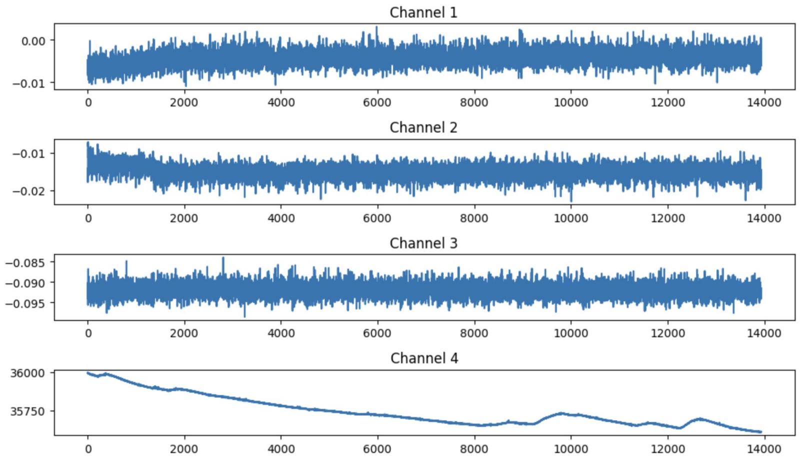
- The dataset measures various emotional dimensions, including Arousal, Valence, Engagement, Liking, Familiarity, using a Likert scale.
- To capture these emotions effectively, multimodal data was recorded, including physiological signals such as EEG (8-channel brain activity at 256 Hz), ECG (heart activity at 256 Hz), and GSR (skin conductance at 8 Hz).
- Additionally, behavioral data such as facial expressions (captured via camera) and head movements (tracked using motion sensors) were collected.
- The dataset is provided in structured files, typically stored in CSV or MATLAB (.mat) formats, containing both raw signals and extracted features for easy processing.

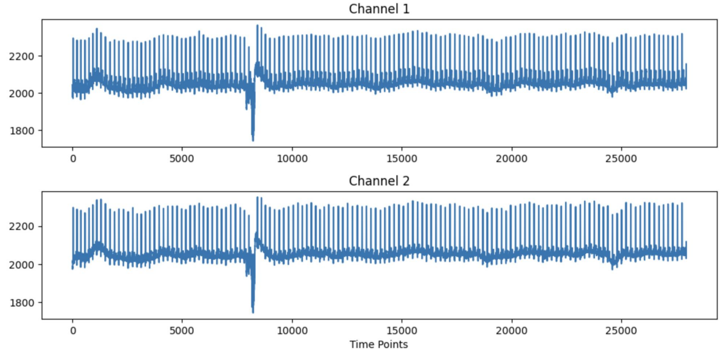
ASCERTAIN Dataset

- **Name**: ASCERTAIN Dataset
- **Purpose**: Emotion and personality trait analysis using EEG and peripheral physiological signals.
- **Dataset Type**: Multimodal physiological dataset (EEG, ECG, EDA/GSR, facial landmarks, etc.)
- **Participants**: 58 subjects (32 males, 26 females)
- **Recording Duration**: Each participant watched a series of 36 videos, with physiological signals recorded throughout each video.
- **Sampling Rate**: EEG and peripheral signals recorded at 1000 Hz (commonly downsampled to 256 Hz or 128 Hz during preprocessing).
- **Total Data Points**: 58 participants × 36 video trials = 2,088 trials

ASCERTAIN Dataset







Fear Data Extraction Process - ASCERTAIN Dataset:

Label	Valence	Arousal	Engagement	Liking	Familiarity
No Fear (0)	0-2	1-3	0-2	5-6	2-4
Low Fear (1)	3-4	-1 to -2	3-4	3-4	1-2
Medium Fear (2)	4-5	-2 to -3	4-5	1-2	0-1
High Fear (3)	5-6	-3	5-6	0-1	0

ASCERTAIN Feature Extraction

- Extract the five features: **Valence, Arousal, Engagement, Liking, Familiarity** from each instance (trial or video).
- For each instance, compare the feature values with the defined range per fear level. Match all five features with one of the fear level rows.
- If all five features fall within a row's value ranges, assign that corresponding fear label (0, 1, 2, or 3).
- EEG signals are decomposed into different frequency bands using band pass filtering - **Theta, Alpha, Low Beta, High Beta and Gamma.**
    - (8 EEG x 5) = **40 features**

ASCERTAIN Feature Extraction

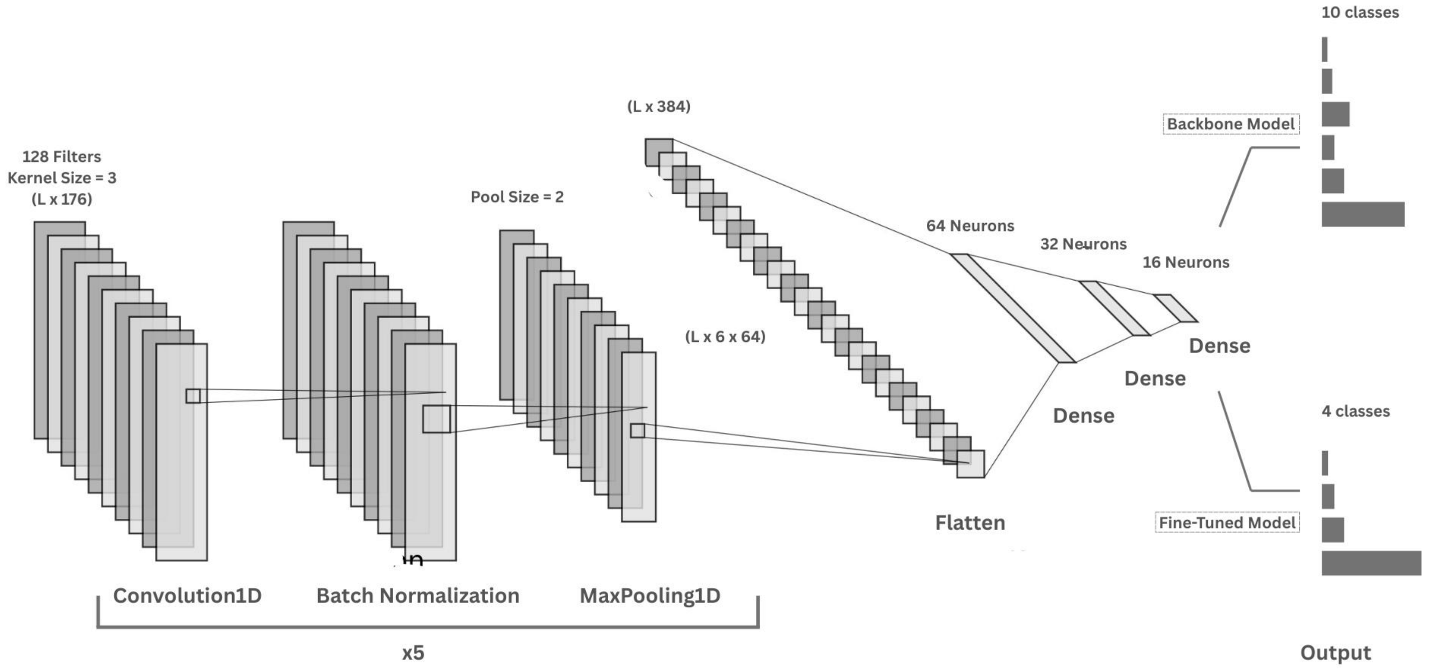
PPS Channels Used

- Selected PPS channels:
  - Accelerometer reading of x, y and z axis
  - GSR - Galvanic Skin Response
  - EEG - for left and right arm

Feature Extraction from PPS Signals

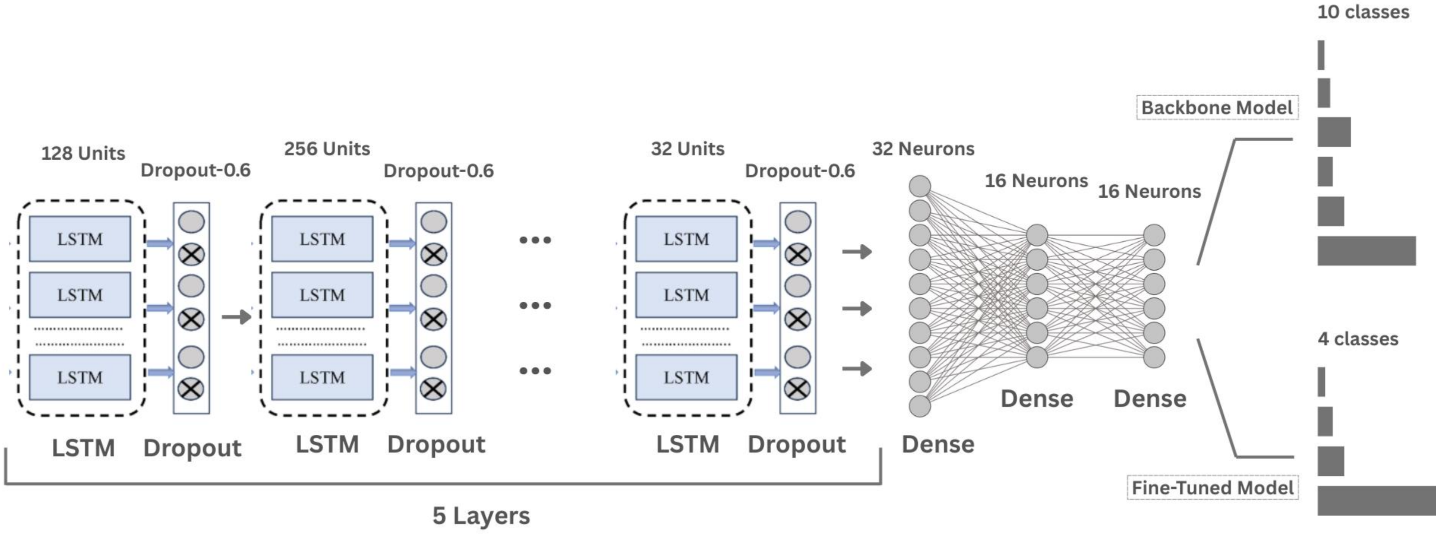
- Extracted statistical features:
  - **Mean (pps_mean)** → np.mean(X)
  - **Variance (pps_var)** → np.var(X)
- Combined size - (8 EEG x 5 bands) + (3 Acc. axis x 2 features) + (1 GSR x 2 features) + (2 EEG x 2 feat.) = **52 features**
- Dependent data, y = n x 7 (Arousal from 1 to 7)

Architecture - CNN Model



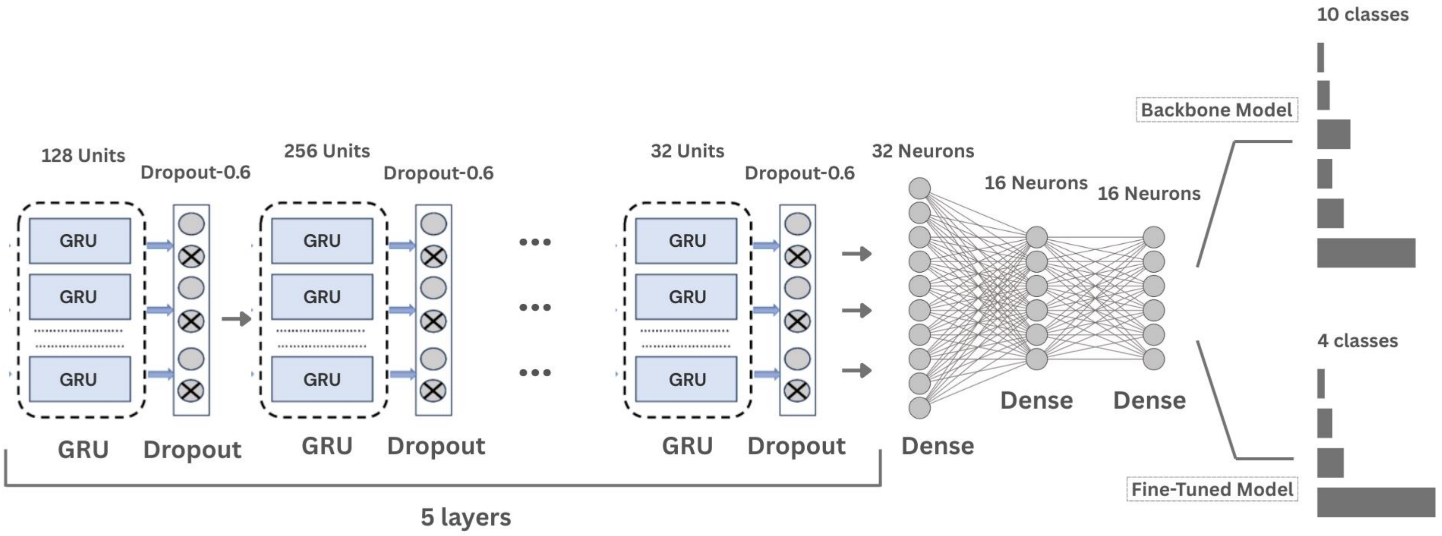
CNN Backbone Model

Architecture - LSTM Model



LSTM Backbone Model

Architecture - GRU Model



GRU Backbone Model

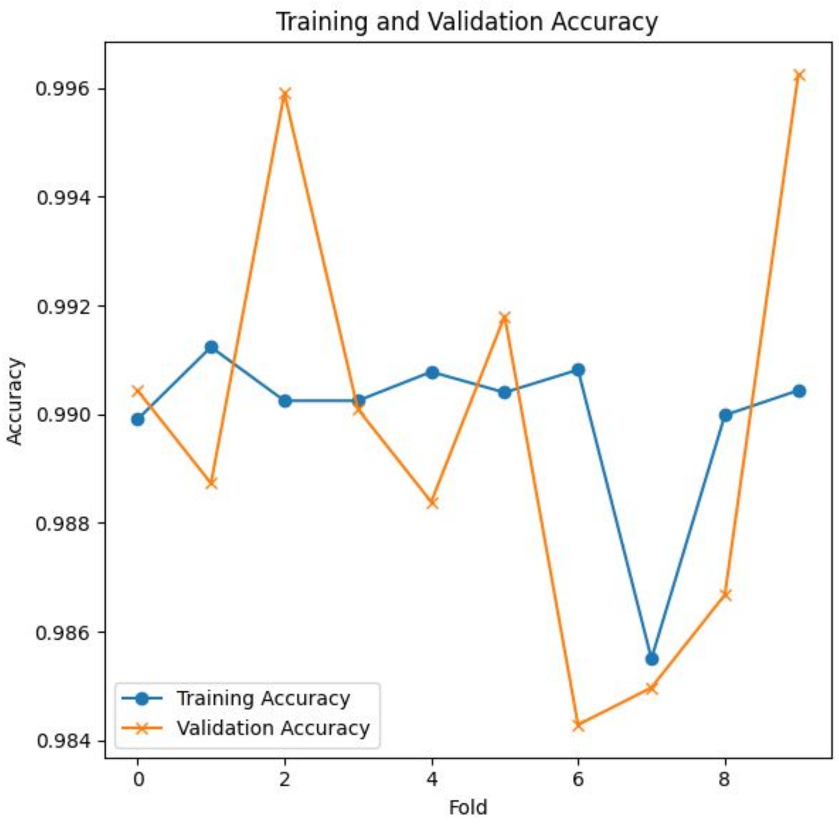
Cross-validation

- **Ensures Generalizability:**
  - Helps evaluate how well the model performs on unseen data across different subjects and sessions.
- **Combats Subject Bias:**
  - Both DEAP and ASCERTAIN datasets have subject-specific variations. Cross-validation prevents the model from overfitting to a specific subject's patterns.
- **Reliable Performance Estimation:**
  - Provides a robust average performance over multiple folds rather than relying on a single train-test split.
- **Maximizes Data Usage:**
  - Especially important with limited participants, as it allows each sample to be used for both training and testing without data leakage.
- **Model Comparison Under Uniform Settings:**
  - Enables fair evaluation and comparison of CNN, LSTM, and GRU by testing them under the same cross-validation conditions.

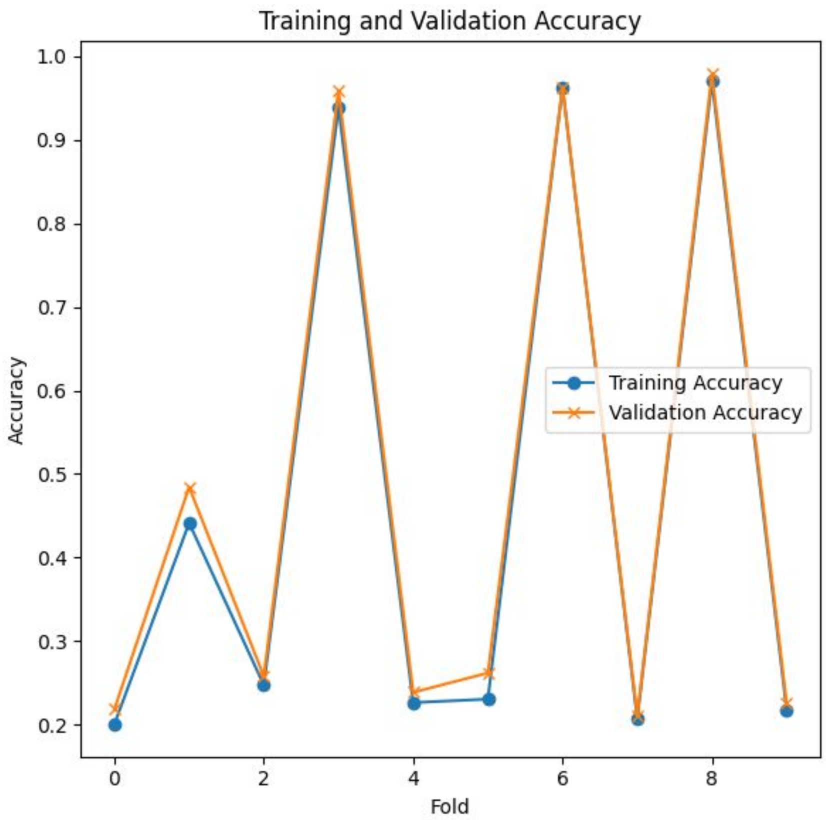
DEAP Dataset: Cross-validation

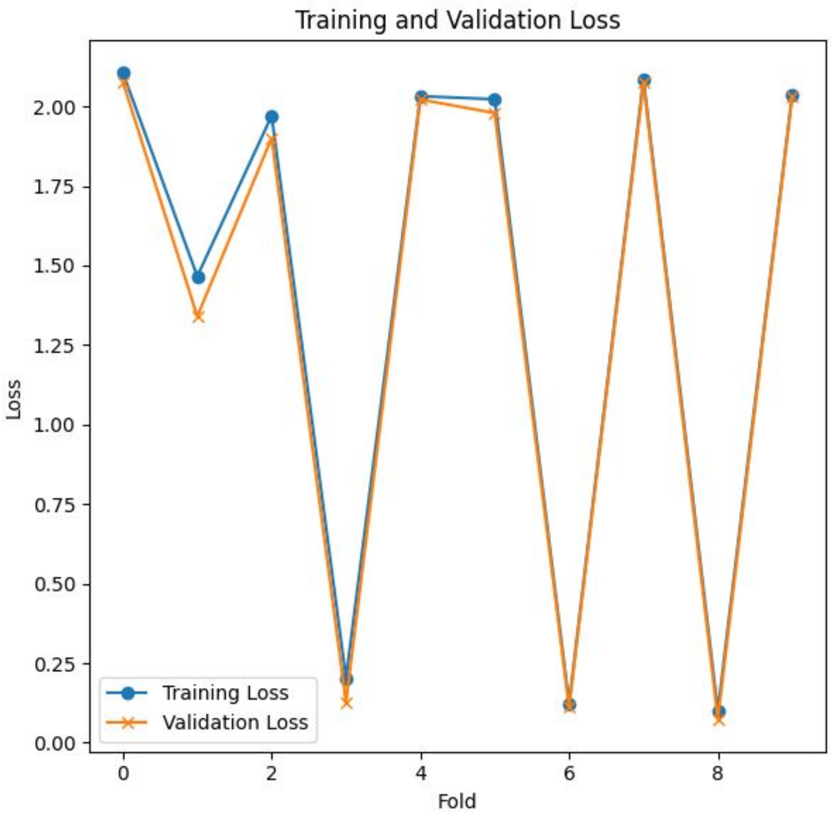
- Used data from **10 subjects of the DEAP** dataset, focusing on self-assessed valence levels with **10 (0 to 9) output classes.**
- Each sample is represented as a 176-dimensional vector, capturing multimodal features(EEG, EOG, GSR) relevant to arousal detection.
- Implemented and tested CNN, LSTM, and GRU models to assess their performance.
- Applied **10-fold cross-validation** to ensure robust evaluation across all 10 subjects (due to resource limitation and high training time), avoiding subject-specific bias.
- The output label shape is n × 7, corresponding to 7 discrete arousal classes derived from the self-assessment manikins provided in the dataset.

DEAP Dataset: Cross-validation CNN

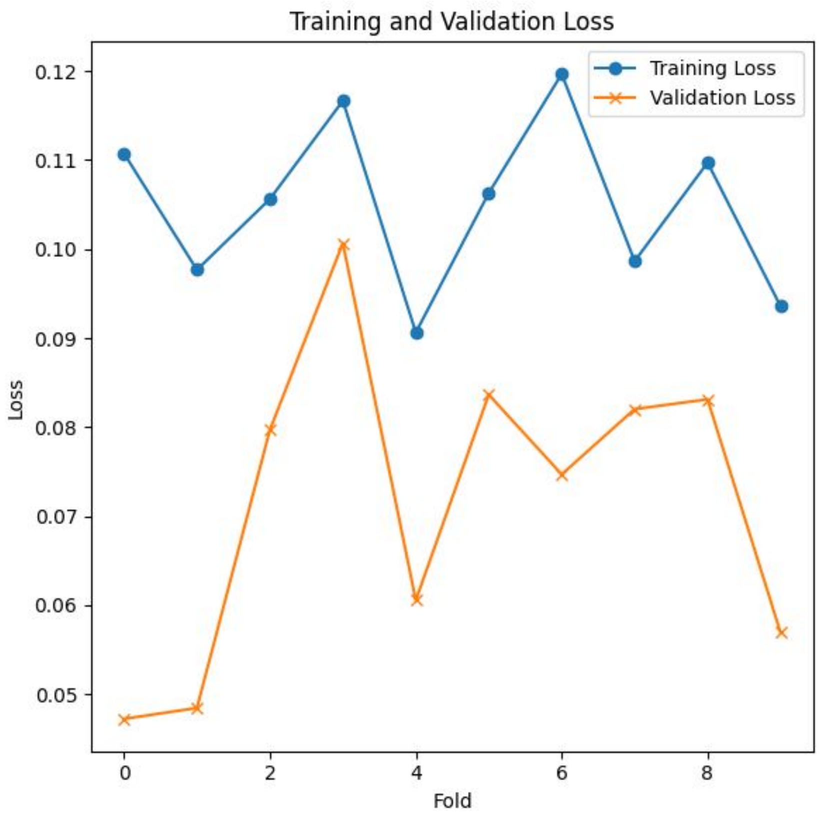


DEAP Dataset: Cross-validation LSTM





DEAP Dataset: Cross-validation GRU



DEAP Dataset: Cross-validation Result

Model	Number of GPUs	Time (in seconds)	Time (in hours)
CNN	2	2049.461	0.56
LSTM	2	9004.959	2.5
GRU	2	8213.382	2.28

- Cross-validation was performed on CNN, LSTM and GRU using Kaggle notebook
- Conducted **dependent two-sample t-test**, where CNN outperformed LSTM and GRU

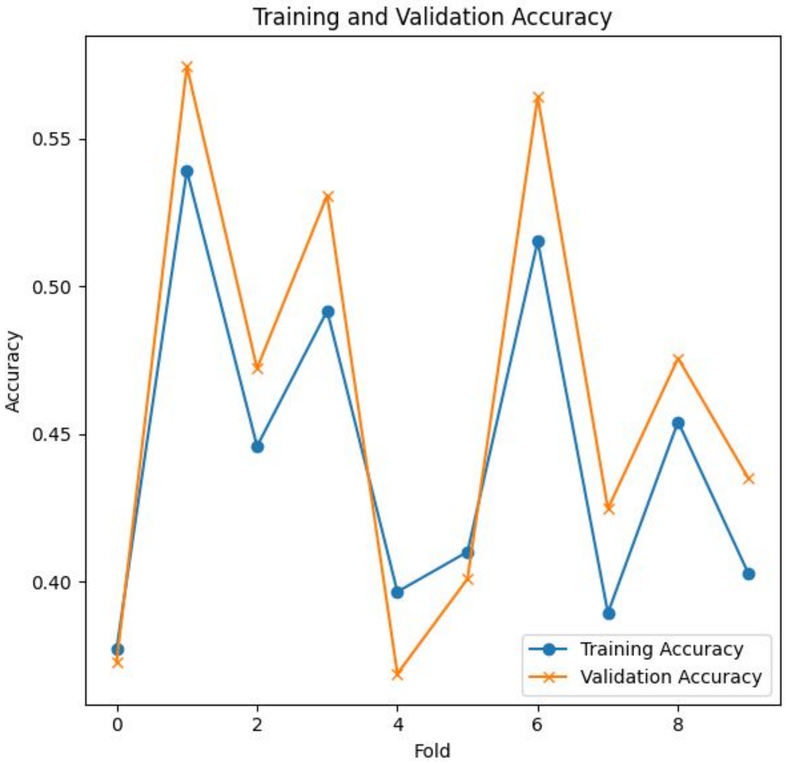
```
T-test between CNN and LSTM (Validation Accuracy): T-statistic = 4.641246745188145 , p-value = 0.0012171110911113274
T-test between CNN and LSTM (Validation Loss): T-statistic = -4.618258574057127 , p-value = 0.0012574572776736406
T-test between CNN and GRU (Validation Accuracy): T-statistic = 4.263906074084673 , p-value = 0.0020990457612394 1
T-test between CNN and GRU (Validation Loss): T-statistic = -4.1909992546044545 , p-value = 0.0023376529001524924
T-test between LSTM and GRU (Validation Accuracy): T-statistic = -4.571459620808026 , p-value = 0.0013440928776451775
T-test between LSTM and GRU (Validation Loss): T-statistic = 4.520777446711981 , p-value = 0.001445181329118955


CNN and LSTM performances are significantly different.
CNN and GRU performances are significantly different.
LSTM and GRU performances are significantly different.
```

ASCERTAIN Dataset: cross-validation

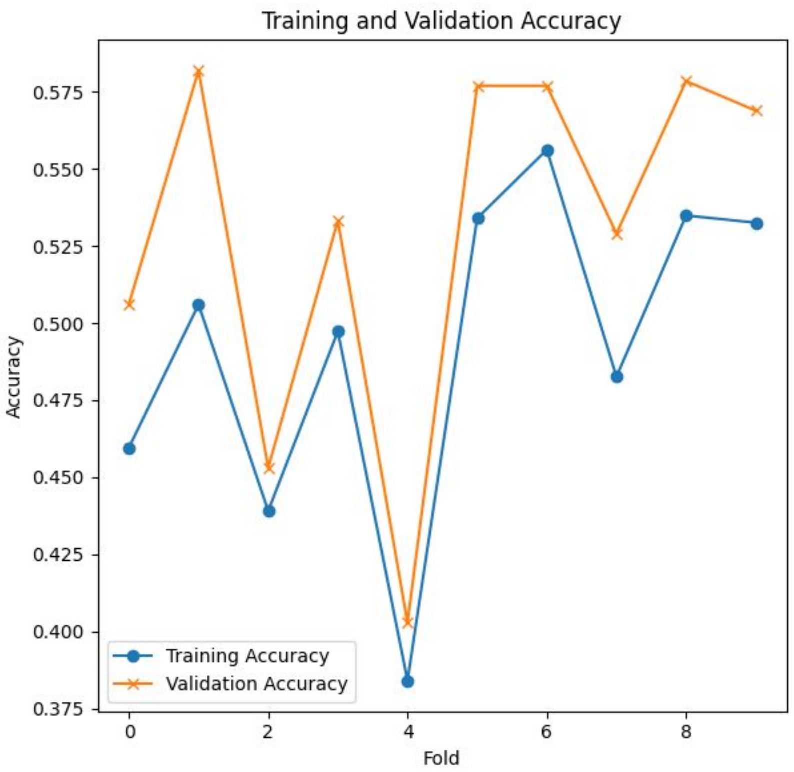
- Used data from **10 subjects of the ASCERTAIN** dataset, focusing on self-assessed arousal levels with 7 output classes.
- Each sample is represented as a 52-dimensional vector, capturing multimodal features(EEG, ECG, GSR) relevant to arousal detection.
- Implemented and tested CNN, LSTM, and GRU models to assess their performance.
- Applied **10-fold cross-validation** to ensure robust evaluation across all 10 subjects (due to resource limitation and high training time), avoiding subject-specific bias.
- The output label shape is $n \times 7$, corresponding to 7 discrete arousal classes derived from the self-assessment manikins provided in the dataset.

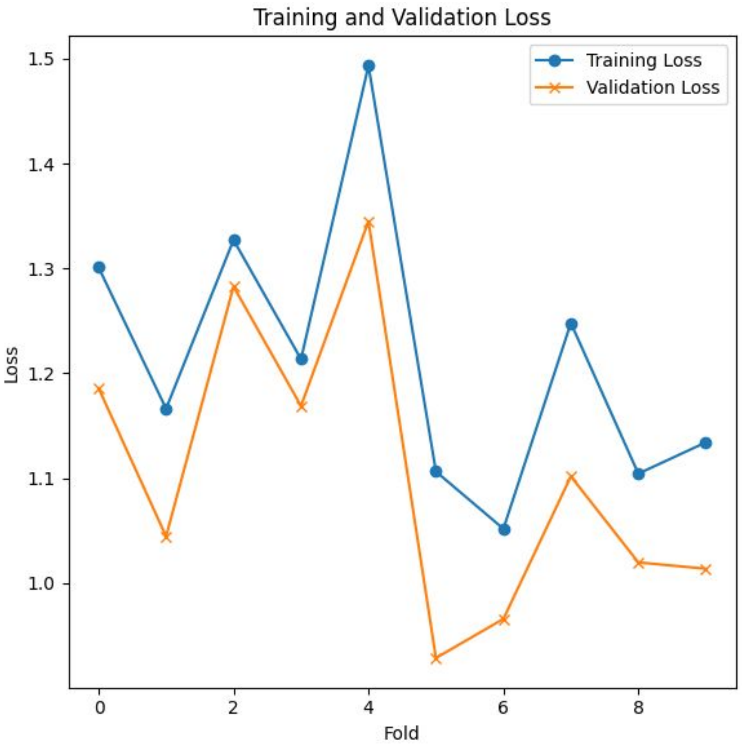
ASCERTAIN Dataset: Cross-validation CNN

ASCERTAIN Dataset: Cross-validation LSTM



ASCERTAIN Dataset: Cross-validation GRU





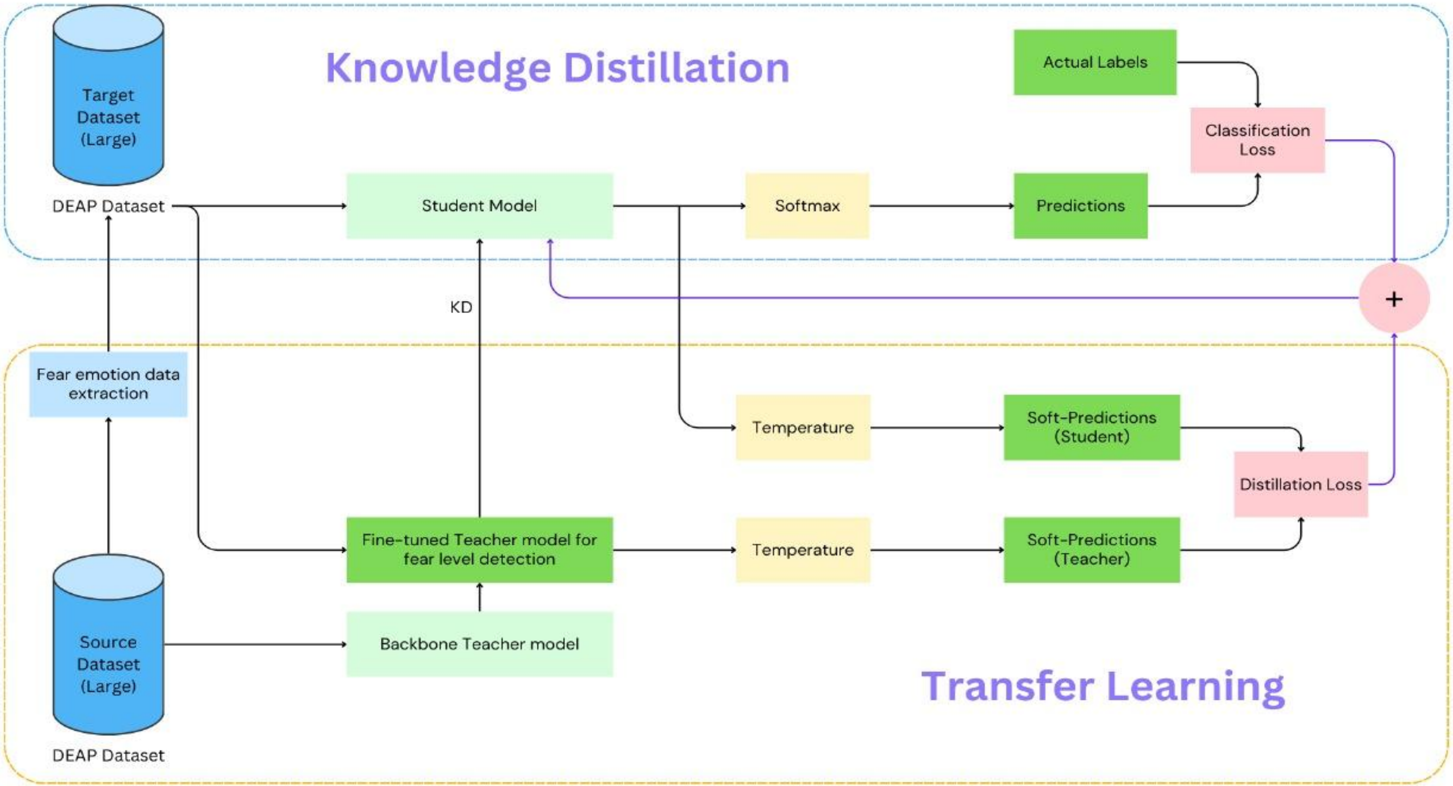
ASCERTAIN Dataset: Cross-validation Result

Model	Number of GPUs	Time (in seconds)	Time (in hours)
CNN	2	1224.933	0.34
LSTM	2	8064.649	2.24
GRU	2	7056.164	1.96

- Cross-validation was performed on CNN, LSTM and GRU using Kaggle notebook
- Conducted **dependent two-sample t-test**, where CNN outperformed LSTM and GRU

```
T-test between CNN and LSTM (Validation Accuracy): T-statistic = 3.8833363297318395 , p-value = 0.00371246937763691
T-test between CNN and LSTM (Validation Loss): T-statistic = -0.843169126972555 , p-value = 0.4209770318742231
T-test between CNN and GRU (Validation Accuracy): T-statistic = 1.7486709749717255 , p-value = 0.11428242898619696
T-test between CNN and GRU (Validation Loss): T-statistic = -0.08093150618950017 , p-value = 0.9372675843853366
T-test between LSTM and GRU (Validation Accuracy): T-statistic = -3.1596446838520893 , p-value = 0.011556821632619561
T-test between LSTM and GRU (Validation Loss): T-statistic = 1.5929869577646143 , p-value = 0.14562716486634295


CNN and LSTM performances are significantly different.
CNN and GRU performances are not significantly different.
LSTM and GRU performances are significantly different.
```

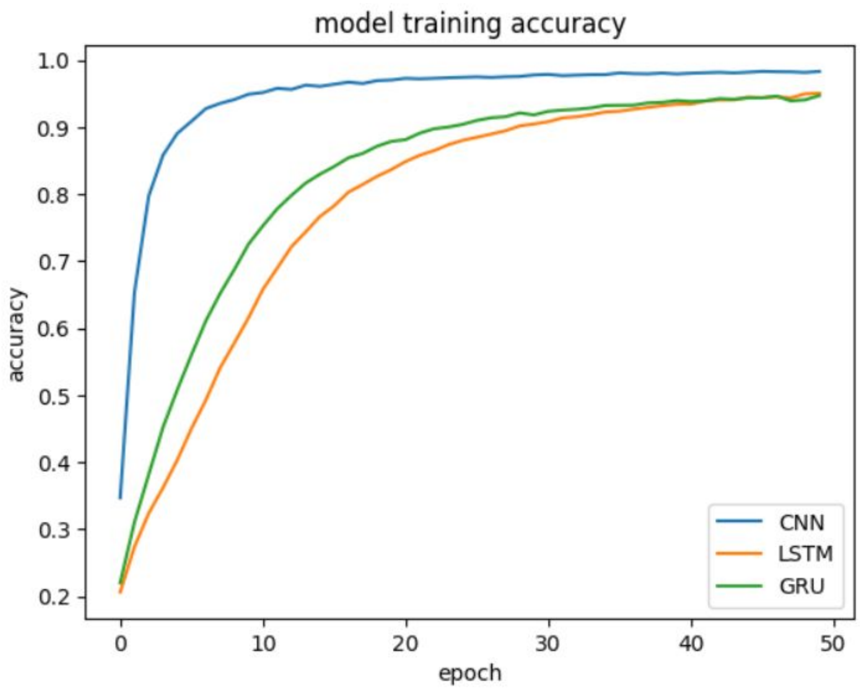
Block Diagram : Method - 1

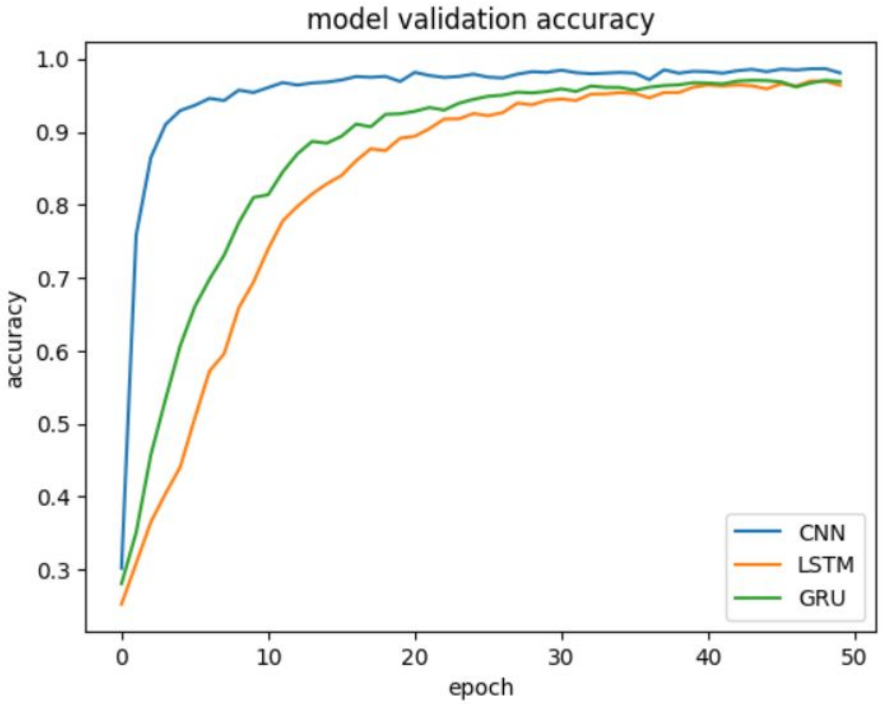


Fine Tuning

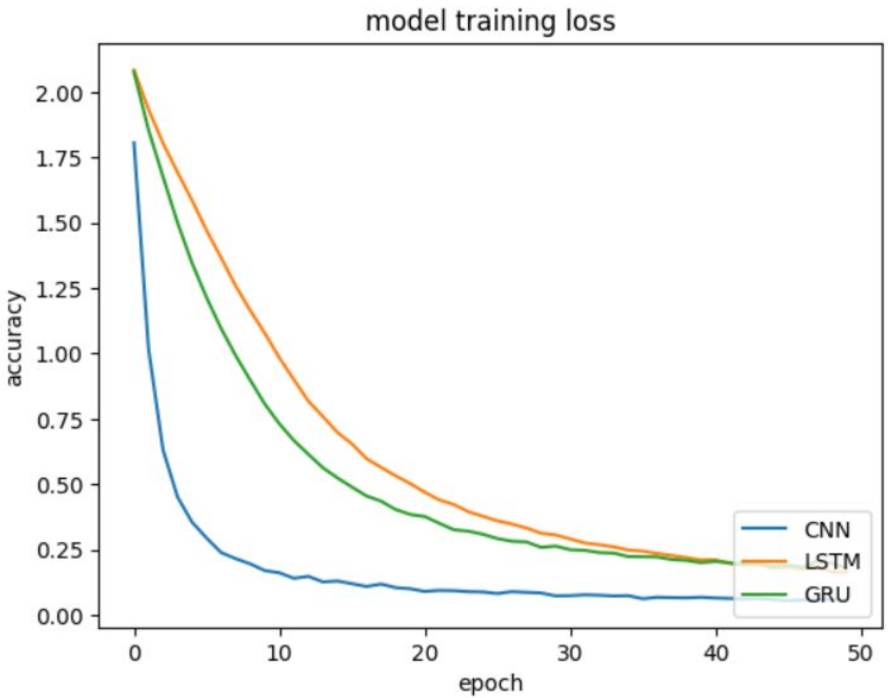
```
input_shape=(fx_train.shape[1], 1) # Define the input shape

new_input = Input(shape=input_shape) # Create an Input layer

# Connect the new input to the first layer of your existing model
x = cnn_model.layers[0](new_input)

# Connect the remaining layers of your existing model
for layer in cnn_model.layers[1:-1]:
    x = layer(x)

# Modify the last layer for 4-class fear detection
new_output = Dense(4, activation='softmax', name='fear_output')(x)

# Create a new model with the modified output layer
fine_tuned_cnn_model = Model(inputs=new_input, outputs=new_output)
```

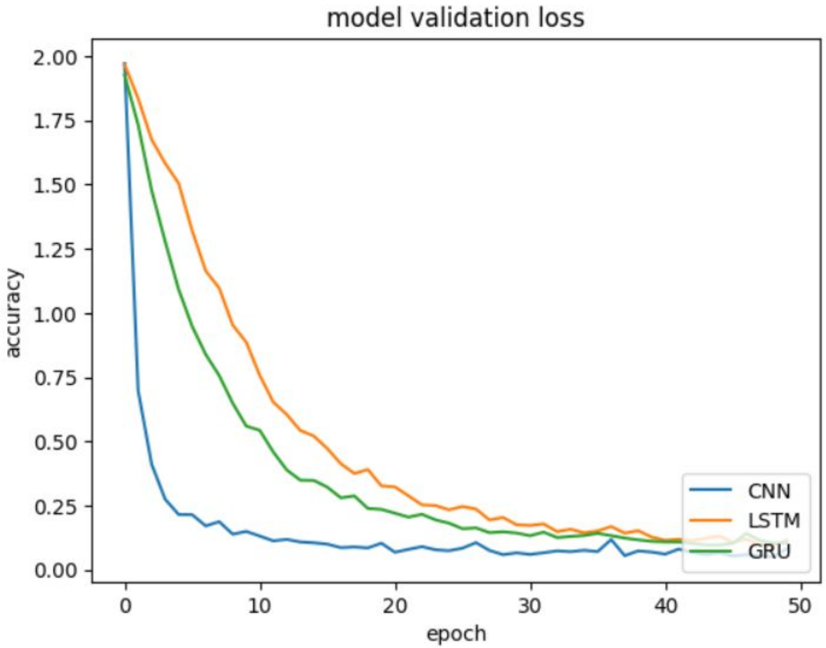
Result Analysis





Result Analysis





Knowledge Distillation

```
stud_cnn_model = Sequential()
input_shape=(r_fx_train.shape[1], 1)
stud_cnn_model.add(Conv1D(128, kernel_size=3,padding = 'same',activation='relu',
                          input_shape=input_shape))
stud_cnn_model.add(BatchNormalization())
stud_cnn_model.add(MaxPooling1D(pool_size=(2)))
stud_cnn_model.add(Conv1D(128,kernel_size=3,padding = 'same', activation='relu'))
stud_cnn_model.add(BatchNormalization())
stud_cnn_model.add(MaxPooling1D(pool_size=(2)))
stud_cnn_model.add(Conv1D(64,kernel_size=3,padding = 'same', activation='relu'))
stud_cnn_model.add(MaxPooling1D(pool_size=(2)))
stud_cnn_model.add(Flatten())
stud_cnn_model.add(Dense(64, activation='tanh'))
stud_cnn_model.add(Dropout(0.2))
stud_cnn_model.add(Dense(16, activation='relu'))
stud_cnn_model.add(Dropout(0.2))
stud_cnn_model.add(Dense(4, activation='softmax'))
```

Teacher Model:

```
Total params: 198,362 (774.85 KB)
Trainable params: 197,338 (770.85 KB)
Non-trainable params: 1,024 (4.00 KB)
```

Student Model:

```
Total params: 117,588 (459.33 KB)
Trainable params: 117,076 (457.33 KB)
Non-trainable params: 512 (2.00 KB)
```

Knowledge Distillation

Hard-Labels(No Knowledge Distillation):

- Training the model directly on the class labels

```
# Train the model with fear data
stud_cnn_model.fit(r_fx_train, fy_train,batch_size=batch_size,epochs=100,verbose=1,
                   validation_data=(r_fx_val, fY_val))
```

Soft-Labels:

- Training the model based on the teacher model’s predictions

```
# Get the teacher model’s predictions
soft_targets = fine_tuned_cnn_model.predict(fx_train)

# Train the student model with soft-targets
stud_cnn_model.fit(r_fx_train, soft_targets,batch_size=batch_size,epochs=100,verbose=1,
                   validation_data=(r_fx_val, fY_val))
```

Knowledge Distillation

Custom Loss Function:

- Combines Distillation loss and Classification loss

```
# Classification loss (student vs ground truth)
classification_loss_fn = tf.keras.losses.CategoricalCrossentropy()
classification_loss = classification_loss_fn(y_true, y_pred_student)

# Teacher predictions for the current batch
y_pred_teacher = teacher_model(X_batch, training=False)
# Distillation loss (student vs teacher predictions)
distillation_loss_fn = tf.keras.losses.KLDivergence()
distillation_loss = distillation_loss_fn(
    tf.nn.softmax(y_pred_teacher / temperature, axis=1),
    tf.nn.softmax(y_pred_student / temperature, axis=1)
)
# Combine losses
combined_loss = alpha * classification_loss + (1 - alpha) * distillation_loss
```

DEAP dataset: Result Analysis

BackBone Model				
Models	Training Accuracy(%)	Training Loss	Test Accuracy(%)	Test Loss
CNN Model	87.33	0.3968	**83.86**	0.4914
LSTM Model	82.96	0.5632	79.58	0.6082
GRU Model	83.02	0.5384	80.43	0.5824

Fine-Tuned Model (for fear data)				
Models	Training Accuracy(%)	Training Loss	Test Accuracy(%)	Test Loss
CNN Model	98.79	0.0379	**99.32**	0.0216
LSTM Model	91.67	0.2067	89.20	0.2581
GRU Model	89.12	0.2702	90.45	0.2385

DEAP dataset: Result Analysis

- Number of channels included for Student model - 14 EEG + 8 PPS = 22 channels
- Total Features extracted = (14 EEG x 5) + (8 PPS x 2) = 86 features

Knowledge Distillation for CNN model				
Targets	Training Accuracy(%)	Training Loss	Test Accuracy(%)	Test Loss
Hard Labels	99.30	0.0356	98.77	0.0388
Soft Targets	99.01	0.0301	98.86	0.0321
Custom Loss Function	98.67	0.0027	**98.96**	**0.0306**

ASCERTAIN dataset: Result Analysis

BackBone CNN model				
Model	Training Accuracy(%)	Training Loss	Test Accuracy(%)	Test Loss
CNN	46.07	1.3305	46.20	1.3372
LSTM	36.93	1.5334	38.67	1.4916
GRU	53.23	1.1597	57.97	1.0272

Fine-tuned model (for fear data)				
Model	Training Accuracy(%)	Training Loss	Test Accuracy(%)	Test Loss
CNN	84.54	0.0888	61.41	80.23
GRU	85.19	0.3290	87.21	0.2972

ASCERTAIN dataset: Result Analysis

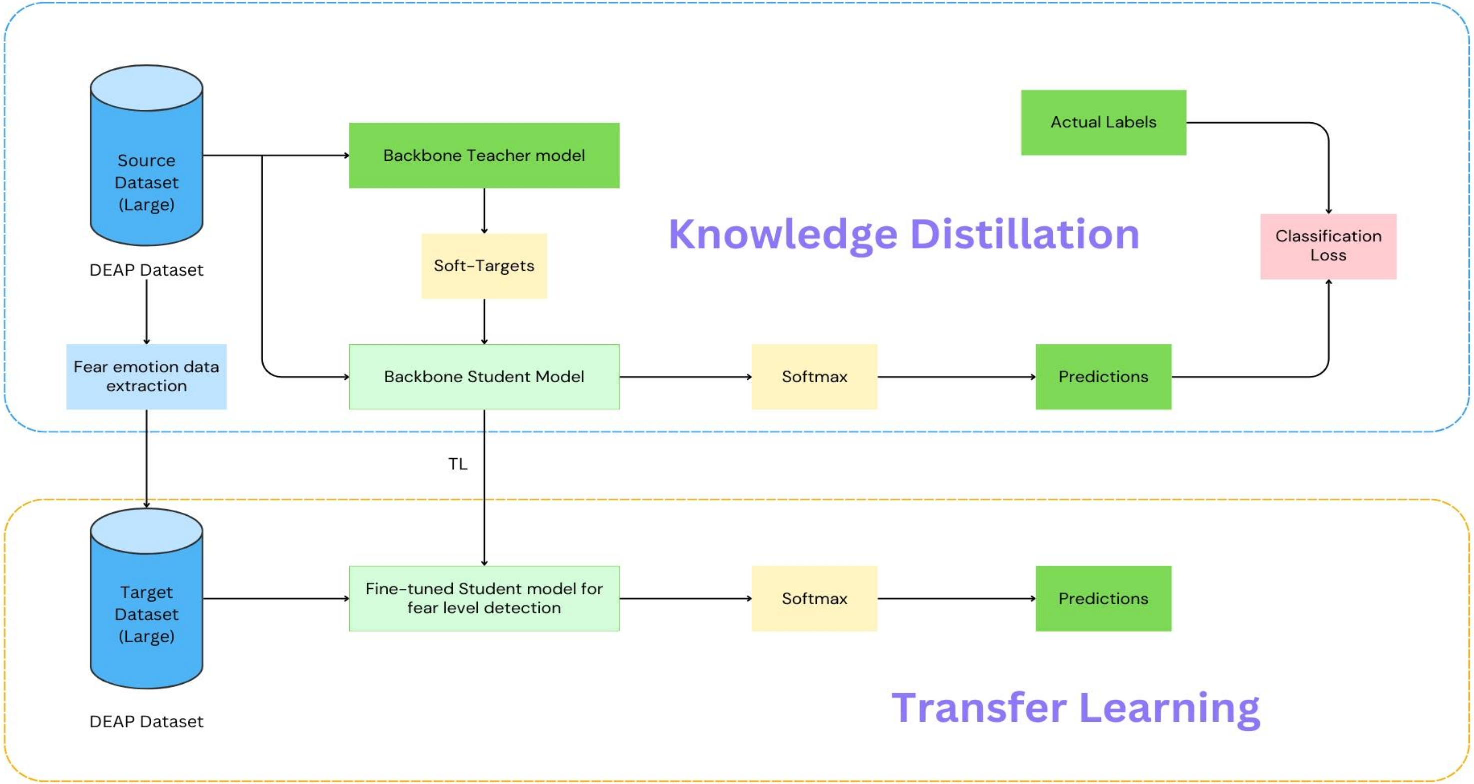
- Number of channels included for Student model - 6 EEG + 1 GSR = 7 channels
- Total Features extracted = (6 EEG x 5) + (1 PPS x 2) = 32 features

Knowledge Distillation CNN Model				
Target	Training Accuracy(%)	Training Loss	Test Accuracy(%)	Test Loss
Hard Labels	46.07	1.3305	46.20	1.3372
Soft Labels	92.85	0.4684	24.69	5.0484
Knowledge Distillation GRU Model				
Hard Labels	65.73	0.7752	70.28	0.6785
Soft Labels	55.63	0.9808	55.97	0.9295

ASCERTAIN and DEAP dataset: Result Analysis

CNN Student Model Trained on both DEAP and ASCERTAIN								
Target	Training Acc.(%)	Training Loss	Test Accuracy(%)					
			Overall		**ASCERTAIN**		**DEAP**	
			Acc.(%)	Loss	Acc. (%)	Loss	Acc.(%)	Loss
Hard Labels	81.57	0.4156	74.25	0.5756	65.93	0.7354	**88.09**	0.3100
Soft Labels	86.34	0.4101	46.78	2.9155	22.62	4.4820	86.95	0.3104
GRU Student Model Trained on both DEAP and ASCERTAIN								
Hard Labels	68.97	0.6856	59.09	0.8780	45.86	1.1391	81.10	0.4439
Soft Labels	77.89	0.5188	69.92	0.6945	58.45	0.9469	**88.98**	0.2748

Block Diagram-Model 2



Result Analysis - Knowledge Distillation for CNN model

Teacher Model - CNN Model				
Model	Training Accuracy(%)	Training Loss	Test Accuracy(%)	Test Loss
CNN	98.67	0.0027	98.96	0.0306

Student Model - CNN Model				
Targets	Training Accuracy(%)	Training Loss	Test Accuracy(%)	Test Loss
Hard Labels	78.45	0.1498	51.89	4.0365
Soft Targets	97.54	0.0706	55.65	3.0219
Custom Loss Function	49.42	0.1021	52.90	1.6003

Result Analysis - Knowledge Distillation for CNN model

Fine Tune - Student Model				
Targets	Training Accuracy(%)	Training Loss	Test Accuracy(%)	Test Loss
Hard Labels	64.65	0.7314	65.79	0.9305
Soft Targets	63.80	0.8305	67.80	0.7738
Custom Loss Function	96.10	0.0032	95.03	0.0740

Method-1 vs Method-2

Aspect	**Method - 1**	**Method - 2**
Training Process	Backbone model is trained, TL is applied followed by KD	Backbone model is trained, KD is applied followed by TL
Data	Source data used to train Backbone model, fear data used to train Teacher and Student model	Source data is used to train backbone and teacher model, fear data is used to train student model.
Loss Function	Combination of Classification loss and Distillation loss	Primarily classification loss for KD and uses soft targets
Teacher Model	Trains with 176 features and performs fear level classification (TL)	Trains with 86 features and performs valence prediction, smaller size (KD)
Student Model	Trained with custom loss function	Trained with Soft-targets from the teacher mode

Requirements

Hardware Requirements

- Processor: Intel i5 series
- RAM: 16 GB/32 GB
- GPU: 4 GB

Software Requirements

- Operating System: Windows 11
- Programming Language: Python 3.8
- Packages: Pytorch, Matplotlib, TensorFlow, Pandas, Scikit-learn, NumPy, Seaborn, Git
- IDE: Google Colaboratory

Dataset

- DEAP- Dataset for Emotional Analysis using Physiological signals.
- ASCERTAIN - Dataset for implicit personality and affect recognition.

Conclusion

- **Method 1 Superiority**: Transfer Learning followed by Knowledge Distillation (Method 1) outperformed Method 2 in terms of training efficiency, accuracy, and stability across all experiments.
- **High Accuracy Achieved**: Fine-tuning the CNN backbone led to near-perfect test accuracy (99.32%), and the student model maintained high performance (up to 98.96%) with reduced complexity.
- **Effective Knowledge Transfer**: Transfer Learning helped the model internalize fear-specific patterns, making subsequent distillation to student models more effective.
- **Method 2 Limitations**: Starting with Knowledge Distillation (Method 2) resulted in weaker early-stage student models and required heavy fine-tuning to achieve decent accuracy, making it less efficient.
- **Cross-Dataset Success**: Method 1 showed strong generalization across DEAP and ASCERTAIN datasets, validating it as a scalable approach for emotion classification in real-world scenarios.

Future Enhancements

- **Cross-Dataset Adaptation**: Introduce domain adaptation, unsupervised transfer learning, or adversarial training to improve performance across different datasets.
- **Attention and Subject Invariance**: Integrate attention mechanisms and subject-invariant feature extraction to enhance focus on key EEG regions and reduce variability across users.
- **Multimodal Input Expansion**: Enrich the feature space by adding facial expressions, speech cues, or eye-tracking data for more comprehensive emotion recognition.
- **Lightweight Architectures**: Explore efficient models like MobileNet, EfficientNet, or transformers for real-time and edge device deployment.
- **Personalized Emotion Tracking**: Develop adaptive systems that adjust classification thresholds using individual baselines for more personalized predictions.

References

- N. Masuda and I. E. Yairi, "Multi-Input CNN-LSTM deep learning model for fear level classification based on EEG and peripheral physiological signals," Frontiers in Psychology, vol. 14, art. no. 1141801, 2023. Frontiers Media. doi: 10.3389/fpsyg.2023.1141801.
- M. Z. I. Ahmed, N. Sinha, E. Ghaderpour, S. Phadikar, and R. Ghosh, "A novel baseline removal paradigm for subject-independent features in emotion classification using EEG," Bioengineering, vol. 10, art. no. 54, 2023. MDPI. doi: 10.3390/bioengineering10010054.
- O. Bălan, G. Moise, A. Moldoveanu, M. Leordeanu, and F. Moldoveanu, "An investigation of various machine and deep learning techniques applied in automatic fear level detection and acrophobia virtual therapy," Sensors, vol. 20, art. no. 496, 2020. MDPI. doi: 10.3390/s20020496.
- A. Dziedzickis, A. Kaklauskas, and V. Bucinskas, "Human emotion recognition: review of sensors and methods," Sensors, vol. 20, art. no. 592, 2020. MDPI. doi: 10.3390/s20030592.
- R. Ghosh, S. Phadikar, N. Deb, N. Sinha, R. Das, and E. Ghaderpour, "Automatic eye-blink and muscular artifact detection and removal from EEG signals using k-nearest neighbour classifier and long short-term memory networks," IEEE Sensors Journal, vol. 23, pp. 5422–5436, 2023. IEEE. doi: 10.1109/JSEN.2023.3237383.

References

- X. Li, S. Chen, X. Hu, and J. Yang, “Understanding the disharmony between dropout and batch normalization by variance shift,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Silver Spring, MD, USA, 2019, pp. 2682–2690. IEEE.
- J. A. Miranda, F. C. Manuel, L. Gutiérrez-Martín, J. M. Lanza-Gutierrez, M. Portela-García, and C. López-Ongil, “Fear recognition for women using a reduced set of physiological signals,” Sensors, vol. 21, no. 5, Art. no. 5, 2021. MDPI. doi: 10.3390/s21051587.
- L. Yang et al., “Edgetb: A hybrid testbed for distributed machine learning at the edge with high fidelity,” IEEE Transactions on Parallel and Distributed Systems, vol. 33, no. 10, 2022. IEEE.
- C. E. Orozco-Mora, D. Oceguera-Cuevas, R. Q. Fuentes-Aguilar, and G. Hernández-Melgarejo, “Stress level estimation based on physiological signals for virtual reality applications,” IEEE Access, vol. 10, pp. 68755–68767, 2022. IEEE. doi: 10.1109/ACCESS.2022.3186318.
- L. Petrescu, C. Petrescu, A. Oprea, O. Mitruț, G. Moise, A. Moldoveanu, et al., “Machine learning methods for fear classification based on physiological features,” Sensors, vol. 21, art. no. 4519, 2021. MDPI. doi: 10.3390/s21134519.


Thank you!