Emotion Recognition using EEG and ECG Data with LSTM and CNN

A detailed explanation of the DREAMER Dataset

Presented by:
FAYROUZ AHMED
YOUSEF GOMAA
MOHAMED ASHRAF
SEIF DIAA

CONTENTS:

INTRODUCTION:

- ✓ Emotion recognition is the process of identifying human emotions.
- ✓ Physiological signals such as EEG (Electroencephalography) and ECG (Electrocardiography) provide objective measures of emotional states.
- ✓ These signals are less prone to conscious control, providing more reliable data.

EMOTIV EPOC Wireless Headset & SHIMMER ECG Device

SHIMMER ECG Device Usage

INTRODUCTION:

- ✓ The DREAMER dataset (.mat) includes EEG and ECG recordings from 23 participants each exposed to 18 stimuli.

Audio-visual stimuli	
Number of videos	18
Video content	Audio-Video
Video duration	65 - 393 s (M=199 s)
Experiment information	
Number of participants	25 (23)
Number of males	14 (14)
Number of females	11 (9)
Age of participants	22 - 33 (M=26.6, SD=2.7)
Rating scales	Arousal, Valence, Dominance
Rating values	1 - 5
Recorded signals	14-channel 128Hz EEG, 256Hz ECG

Field	Value
Data	1x23 cell
EEG_SamplingRate	128
ECG_SamplingRate	256
EEG_Electrodes	1x14 cell
noOfSubjects	23
noOfVideoSequen...	18
Disclaimer	'While every care has been taken to ensure the accuracy of the data included in the DREAMER dataset, the authors and the Universi...' ...
Provider	'University of the West of Scotland'
Version	'1.0.2'
Acknowledgement	'The authors would like to thank Thomas Cuntz and Sebastian Palke for the data collection under their BSc (Hons) project.'

Variables - DREAMER.Data{1, 22}

+3 DREAMER DREAMER.EEG_El

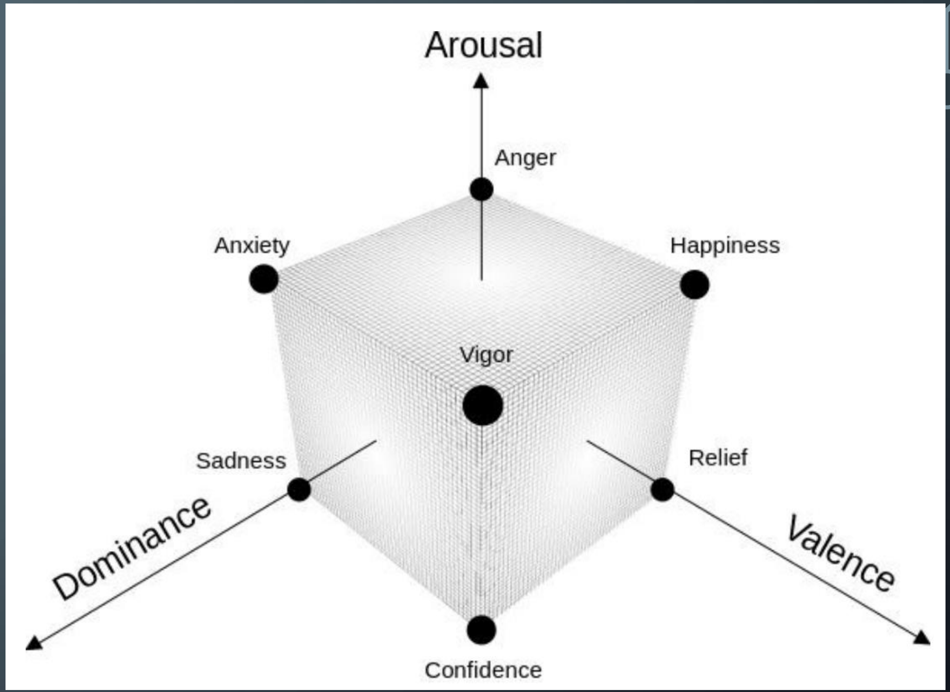
DREAMER.Data{1, 22}

Field	Value
Age	'28'
Gender	'male'
EEG	1x1 struct
ECG	1x1 struct
ScoreValence	18x1 double
ScoreArousal	18x1 double
ScoreDominance	18x1 double

- ✓ The 14 Electrodes: AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8 and AF4.

EEG AND ECG SIGNALS:

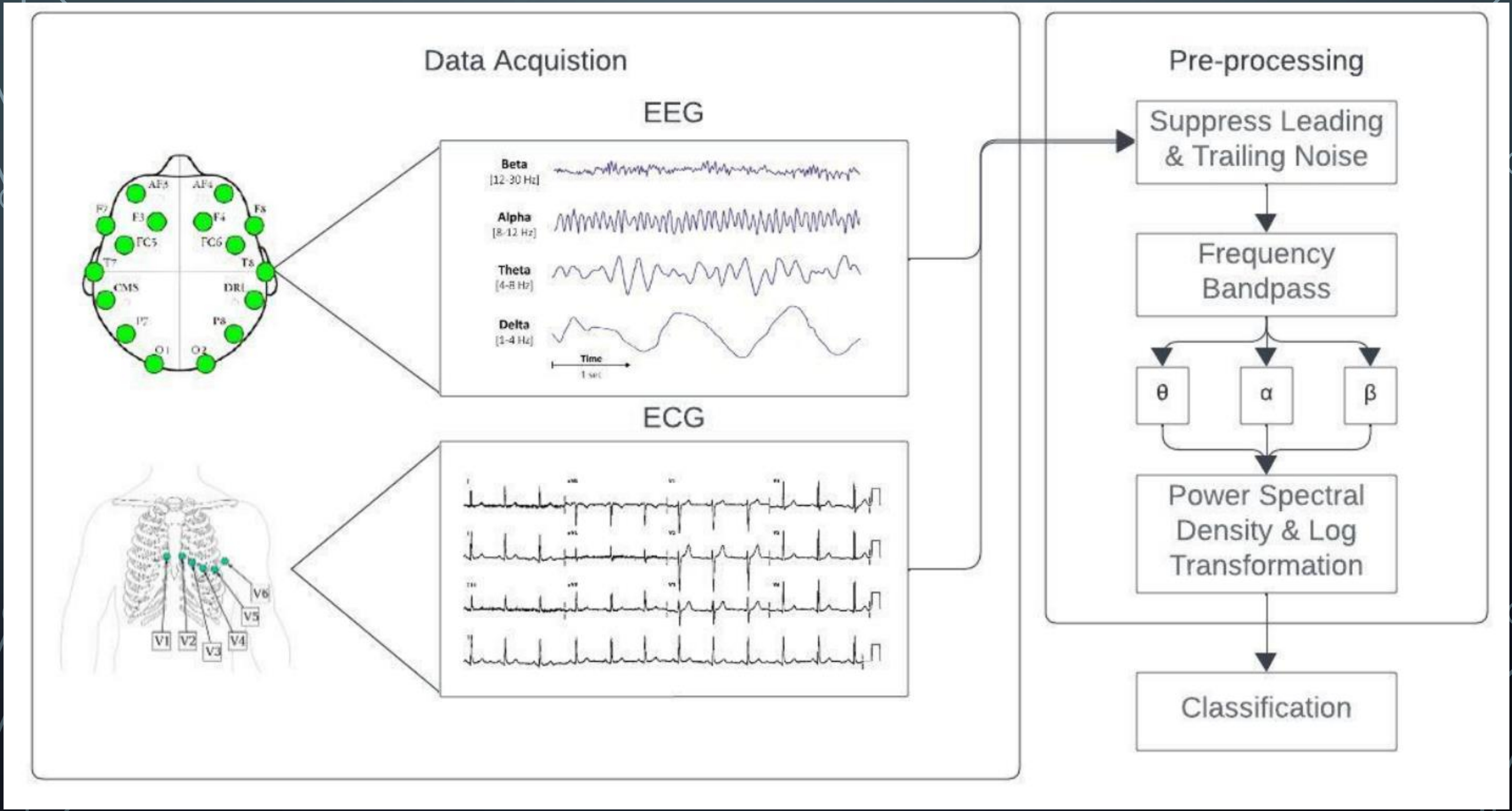
- ✓ EEG measures brain activity through electrodes placed on the scalp.
- ✓ ECG measures heart activity, providing information about heart rate and rhythm.



EMOTION LABELS

- ✓ DREAMER captures emotional responses to audio-visual stimuli, labeled as Arousal, Valence, Dominance.

- ✓ Arousal: Measures the intensity of the emotion (calm to excited).
- ✓ Valence: Measures the positivity or negativity of the emotion.
- ✓ Dominance: Measures the feeling of control in the emotion.

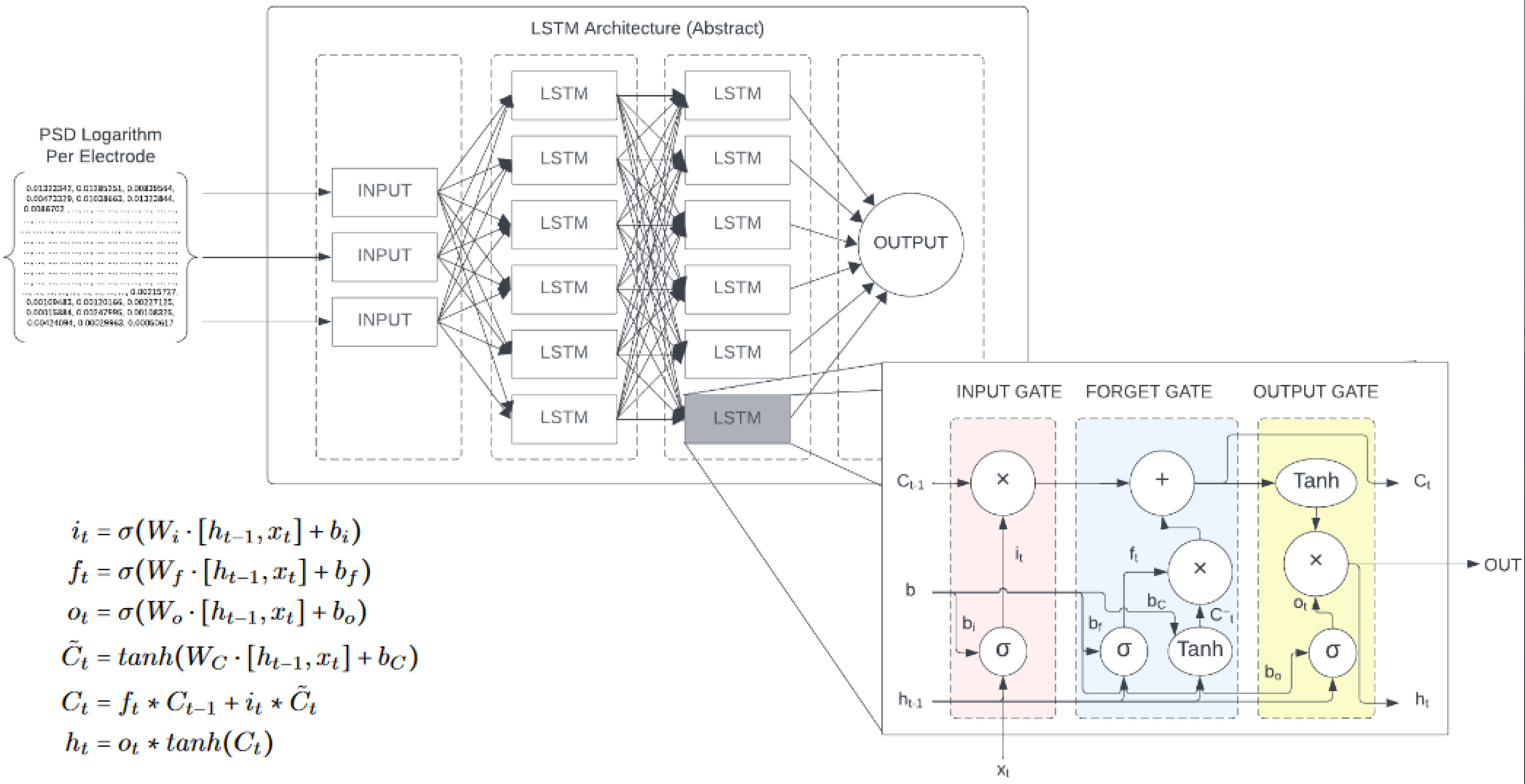

Data Acquistion
EEG
AF3
AF4
F7
F8
F3
F4
FC5
FC6
T7
T8
CMS
DRL
P7
P8
O1
O2
Beta
[12-30 Hz]
Alpha
[8-12 Hz]
Theta
[4-8 Hz]
Delta
[1-4 Hz]
Time
ECG
V1
V2
V3
V4
V5
V6
Pre-processing
Suppress Leading & Trailing Noise
Frequency Bandpass
θ
α
β
Power Spectral Density & Log Transformation
Classification

DATA PREPROCESSING & EXTRACTION

```python
if __name__ == '__main__':
    total=0
    path=u'DREAMER.mat'
    data=sio.loadmat(path)
    print("ECG signals are being feature extracted...")
    ECG={}
    for k in range(0,23):
        for j in range(0,18):
            basl_l=data['DREAMER'][0,0]['Data'][0,k]['ECG'][0,0]['baseline'][0,0][j,0][:,0]
            stim_l=data['DREAMER'][0,0]['Data'][0,k]['ECG'][0,0]['stimuli'][0,0][j,0][:,0]
            basl_r=data['DREAMER'][0,0]['Data'][0,k]['ECG'][0,0]['baseline'][0,0][j,0][:,1]
            stim_r=data['DREAMER'][0,0]['Data'][0,k]['ECG'][0,0]['stimuli'][0,0][j,0][:,1]
            ecg_signals_b_l,info_b_l=nk.ecg_process(basl_l,sampling_rate=256)
            ecg_signals_s_l,info_s_l=nk.ecg_process(stim_l,sampling_rate=256)
            ecg_signals_b_r,info_b_r=nk.ecg_process(basl_r,sampling_rate=256)
            ecg_signals_s_r,info_s_r=nk.ecg_process(stim_r,sampling_rate=256)
            # processed_ecg_b_l = nk.ecg_intervalrelated(ecg_signals_b_l)
            # processed_ecg_s_l = nk.ecg_intervalrelated(ecg_signals_s_l)
            # processed_ecg_b_r = nk.ecg_intervalrelated(ecg_signals_b_r)
            # processed_ecg_s_r = nk.ecg_intervalrelated(ecg_signals_s_r)
            processed_ecg_l=nk.ecg_intervalrelated(ecg_signals_s_l)/nk.ecg_intervalrelated(ecg_signals_b_l)
            processed_ecg_r=nk.ecg_intervalrelated(ecg_signals_s_r)/nk.ecg_intervalrelated(ecg_signals_b_r)
            processed_ecg=(processed_ecg_l+processed_ecg_r)/2
            if not len(ECG):
                ECG=processed_ecg
            else:
                ECG=pd.concat([ECG,processed_ecg],ignore_index=True)
            total+=1
            print("\rprogress: %d%%" %(total/(23*18)*100),end="")
    # col=ECG.columns.values
    # scaler=pre.StandardScaler()
    # for i in range(len(col)):
    #     ECG[col[i][:-3]] = scaler.fit_transform(ECG[[col[i]]])
    # ECG.drop(col, axis=1, inplace=True)
```

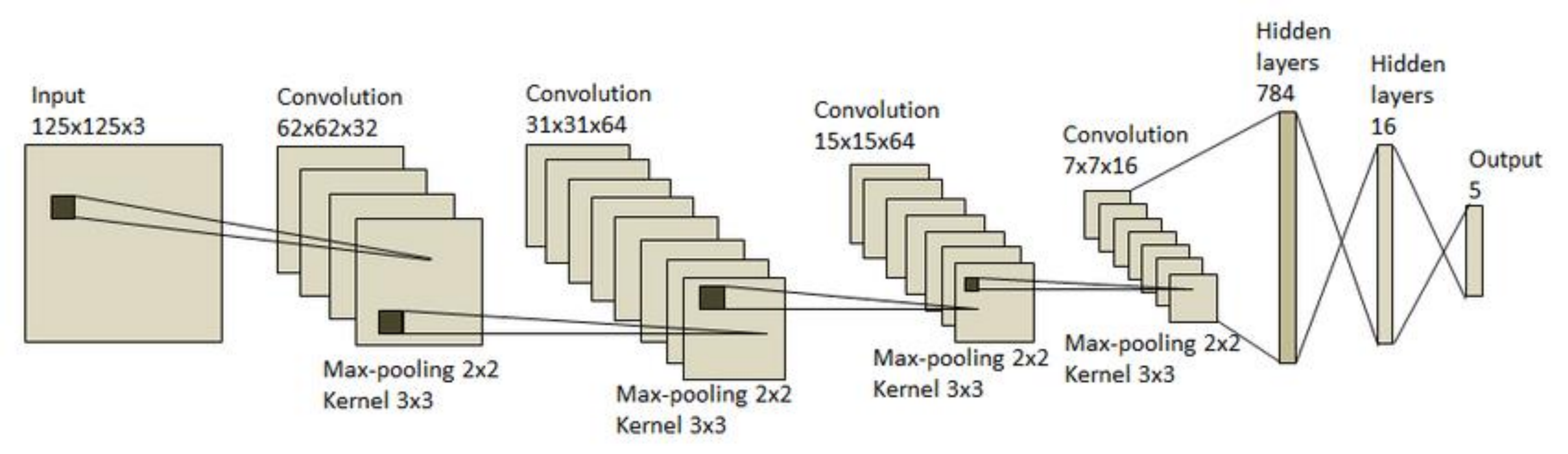
```python
def preprocessing(input,feature):
    overall=signal.firwin(9,[0.0625,0.46875],window='hamming')
    theta=signal.firwin(9,[0.0625,0.125],window='hamming')
    alpha=signal.firwin(9,[0.125,0.203125],window='hamming')
    beta=signal.firwin(9,[0.203125,0.46875],window='hamming')
    filtedData=signal.filtfilt(overall,1,input)
    filtedtheta=signal.filtfilt(theta,1,filtedData)
    filtedalpha=signal.filtfilt(alpha,1,filtedData)
    filtedbeta=signal.filtfilt(beta,1,filtedData)
    ftheta,psdtheta=signal.welch(filtedtheta,nperseg=256)
    falpha,psdalpha=signal.welch(filtedalpha,nperseg=256)
    fbeta,psdbeta=signal.welch(filtedbeta,nperseg=256)
    feature.append(max(psdtheta))
    feature.append(max(psdalpha))
    feature.append(max(psdbeta))
    return feature


if __name__ == '__main__':
    total=0
    path=u'DREAMER.mat'
    data=sio.loadmat(path)
    print("EEG signals are being feature extracted...")
    EEG_tmp=np.zeros((23,18,42))
    for k in range(0,23):
        for j in range(0,18):
            for i in range(0,14):
                B,S=[],[]
                basl=data['DREAMER'][0,0]['Data'][0,k]['EEG'][0,0]['baseline'][0,0][j,0][:,i]
                stim=data['DREAMER'][0,0]['Data'][0,k]['EEG'][0,0]['stimuli'][0,0][j,0][:,i]
                B=preprocessing(basl,B)
                S=preprocessing(stim,S)
                Extrod=np.divide(S,B)
                total+=1
                EEG_tmp[k,j,3*i]=Extrod[0]
                EEG_tmp[k,j,3*i+1]=Extrod[1]
```

MODEL ARCHITECTURE:



$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$
$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$
$$\tilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$
$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$
$$h_t = o_t * tanh(C_t)$$

MODEL ARCHITECTURE:



$$I(x,y) * F(x,y) = \sum_{0}^{M} \sum_{0}^{N} I(n1,n2) \cdot F(x-n1, y-n2)$$

$$ReLU(x) = max(0,x)$$

$$softmax(z_i) = \frac{e^{z_j}}{\sum_j e^{z_j}}$$

MODEL TRAINING AND EVALUATION:

```
models = [
    # NN
    MLPClassifier(activation='logistic',hidden_layer_sizes=(100,3),random_state=7),
    # SVM
    svm.SVC(C=1,random_state=7),
    # DT
    tree.DecisionTreeClassifier(criterion='entropy',max_depth=8,min_samples_leaf=2,min_samples_split=5,random_state=7),
    # GBDT
    GBDT(learning_rate=0.1,max_depth=9,min_samples_leaf=60,min_samples_split=10,n_estimators=31,random_state=7),
    # ada
    ada(LogisticRegression(penalty='l2',C=0.55,max_iter=1000),learning_rate=0.3,n_estimators=4,random_state=7),
    # LR
    LogisticRegression(penalty='l2',C=0.55,max_iter=1000)
]

features = {'Valence':feature_V, 'Arousal':feature_A, 'Dominance':feature_D}

for model in models:
    print("Model:", model.__class__.__name__)
    for feature, data in features.items():
        X_train, X_test, Y_train, Y_test = train_test_split(feature_X, data, test_size=0.3, random_state=7)
        model.fit(X_train, Y_train)
        prediction = model.predict(X_test)
        accuracy = metrics.accuracy_score(Y_test, prediction)
        f1 = metrics.f1_score(Y_test, prediction, average='weighted')
        print(feature, "Accuracy:", accuracy, '\n', feature, "F1:", f1, '\n')
    print("-----------------------------------------------")
```

```
[Running] python -u "e:\GitHub\EmotionDetection-Project
Model: MLPClassifier
Valence Accuracy: 0.568
 Valence F1: 0.516776769509982

Arousal Accuracy: 0.576
 Arousal F1: 0.566522972354748

Dominance Accuracy: 0.608
 Dominance F1: 0.6043271520593403

-----------------------------------------------
Model: SVC
Valence Accuracy: 0.568
 Valence F1: 0.41151020408163264

Arousal Accuracy: 0.496
 Arousal F1: 0.3288983957219251

Dominance Accuracy: 0.472
 Dominance F1: 0.302695652173913

-----------------------------------------------
Model: DecisionTreeClassifier
Valence Accuracy: 0.576
 Valence F1: 0.5620612286829669

Arousal Accuracy: 0.576
 Arousal F1: 0.5729958132045089

Dominance Accuracy: 0.584
 Dominance F1: 0.5843196721311475

-----------------------------------------------
```

MODEL TRAINING AND EVALUATION:

```
--------------------------------------------------
Model: GradientBoostingClassifier
Valence Accuracy: 0.512
 Valence F1: 0.43528082461247936

Arousal Accuracy: 0.608
 Arousal F1: 0.5865523886530557

Dominance Accuracy: 0.568
 Dominance F1: 0.5671700819672131

--------------------------------------------------
Model: AdaBoostClassifier
Valence Accuracy: 0.568
 Valence F1: 0.4253254786450663

Arousal Accuracy: 0.56
 Arousal F1: 0.4745142857142857

Dominance Accuracy: 0.6
 Dominance F1: 0.5753490870032223

--------------------------------------------------
Model: LogisticRegression
Valence Accuracy: 0.544
 Valence F1: 0.4555595791534133

Arousal Accuracy: 0.576
 Arousal F1: 0.5560891029500301

Dominance Accuracy: 0.584
 Dominance F1: 0.5832008196721311

--------------------------------------------------
```

MODEL TRAINING AND EVALUATION:

Model	Modality	Accuracy			F1 Score		
		Valence	Arousal	Dominance	Valence	Arousal	Dominance
CNN	EEG	**0.6249**	0.6217	**0.6184**	0.5184	0.5767	0.6166
	ECG	0.6237	0.6237	0.6157	**0.5305**	**0.5798**	0.6145
	Fusion (EEG & ECG)	0.6184	**0.6232**	0.6184	0.5213	0.5750	**0.6171**
	Random	0.5000	0.5000	0.5000	0.4878	0.4878	0.4895
	Class ratio	0.5440	0.5467	0.5403	0.5000	0.5000	0.5000
LSTM	EEG	**0.6123**	0.5834	**0.6678**	0.5423	0.5912	0.6144
	ECG	0.5637	0.6235	0.5957	**0.5504**	**0.5792**	0.6065
	Fusion (EEG & ECG)	0.5764	**0.6412**	0.5783	0.5217	0.5550	**0.6171**
	Random	0.5000	0.5000	0.5000	0.4878	0.4878	0.4895
	Class ratio	0.5440	0.5467	0.5403	0.5000	0.5000	0.5000

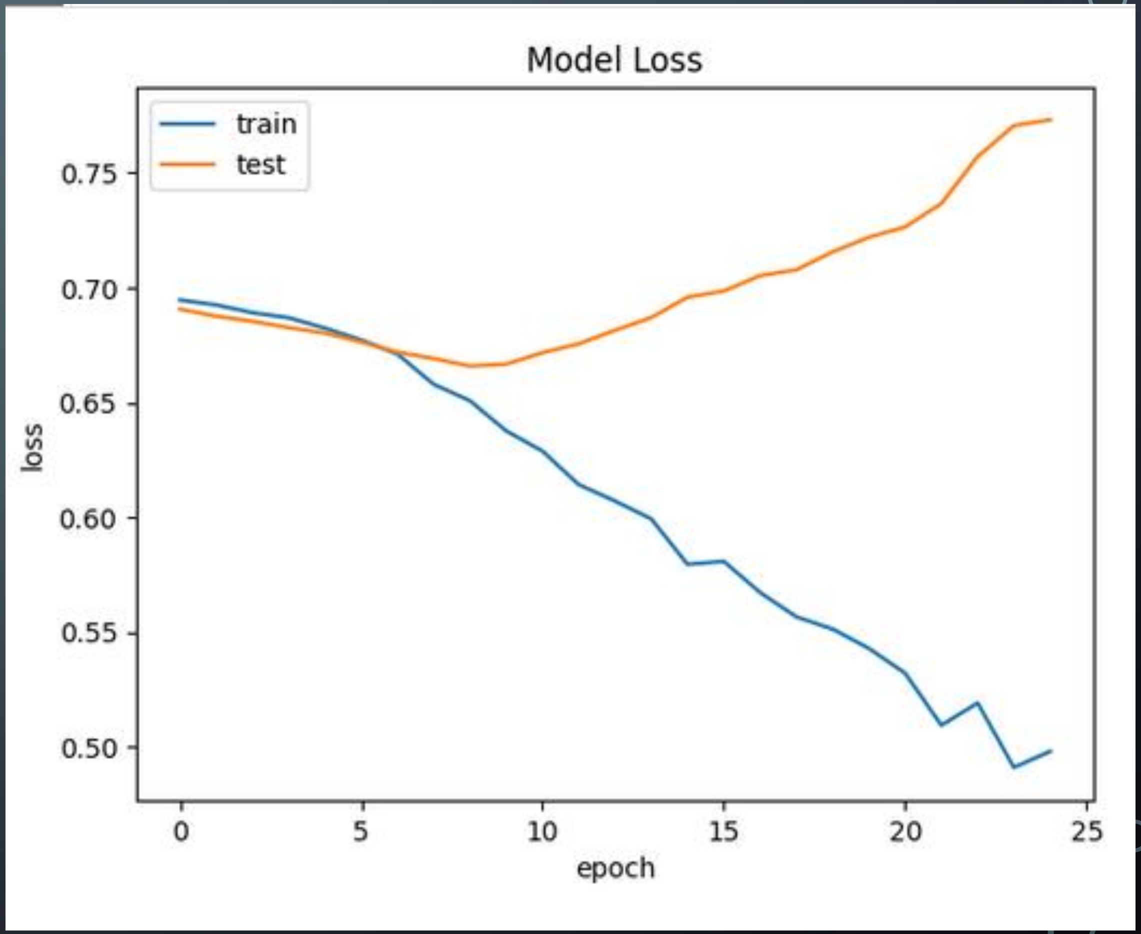
Table 1: Table showing Accuracy and F1 Score for different models and modalities

RESULTS: GROUPED DATASET (LSTM) VERSION 4

✓ In total, there are around 11 million data points, which belong to 414 chunks when grouped by participants & videos (414 == 23*18)

✓ Overfitting due to small dataset(?)

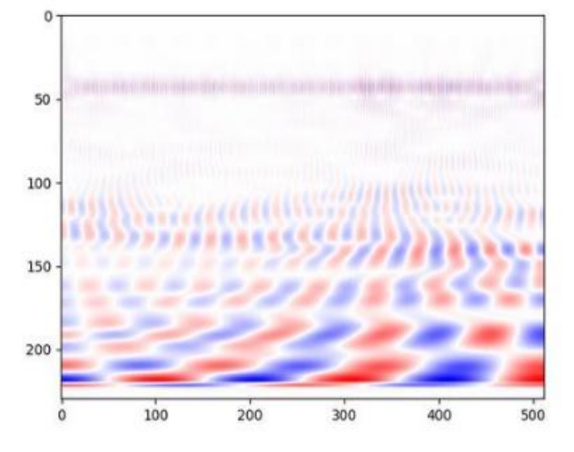
```
In [28]:
score=model3.evaluate(X_test,y_test)
print(f"Accuracy: {score[1]}",f"Loss: {score[0]}")

3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.6279 - loss: 0.7692
Accuracy: 0.6385542154312134 Loss: 0.7729015350341797
```



RESULTS: ORIGINAL DATASET (CNN) VERSION 10

Layer (type)	Output Shape	Param #
conv2d_12 (Conv2D)	(None, 50430, 12, 32)	320
batch_normalization_11 (BatchNormalization)	(None, 50430, 12, 32)	128
max_pooling2d_11 (MaxPooling2D)	(None, 25215, 6, 32)	0
conv2d_13 (Conv2D)	(None, 25213, 4, 64)	18,496
batch_normalization_12 (BatchNormalization)	(None, 25213, 4, 64)	256
max_pooling2d_12 (MaxPooling2D)	(None, 12606, 2, 64)	0
flatten_3 (Flatten)	(None, 1613568)	0
dense_6 (Dense)	(None, 16)	25,817,104
dropout_3 (Dropout)	(None, 16)	0
dense_7 (Dense)	(None, 5)	85

- ✓ 11 million data points.
  - ✓ Insufficient resources (RAM/GPU)
  - ✓ Gigabytes of 2d-arrays (images)




```
_, accuracy_valence = model_valence.evaluate(X_test, y_test_valence)
print('Valence Accuracy:', accuracy_valence)

_, accuracy_arousal = model_arousal.evaluate(X_test, y_test_arousal)
print('Arousal Accuracy:', accuracy_arousal)

_, accuracy_dominance = model_dominance.evaluate(X_test, y_test_dominance)
print('Dominance Accuracy:', accuracy_dominance)
```

```
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 61ms/step - accuracy: 0.6536 - loss: 0.3327
Valence Accuracy: 0.6829314039019407
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 61ms/step - accuracy: 0.6771 - loss: 0.3543
Arousal Accuracy: 0.6650602459907532
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 62ms/step - accuracy: 0.6044 - loss: 0.2931
Dominance Accuracy: 0.6752610452314613
```

RESULTS: ORIGINAL DATASET (TORCHEEG + PYTORCH TSCEPTION) VERSION 11

TABLE 2
Structure of the Proposed TSception

Model structure		Layers	Input	Output
Block1	3 branches (in parallel)	Conv2d, LK-ReLU, AP((1,8)) Kernel=15@(1, 64)	(-1, 1, 28, 512)	(-1, 15, 28, 56)
		Conv2d, LK-ReLU, AP((1,8)) Kernel=15@(1, 32)	(-1, 1, 28, 512)	(-1, 15, 28, 60)
		Conv2d, LK-ReLU, AP((1,8)) Kernel=15@(1, 16)	(-1, 1, 28, 512)	(-1, 15, 28, 62)
		Concatenate, BN		(-1, 15, 28, 178)
Block2	2 branches (in parallel)	Conv2d, LK-ReLU, AP((1,2)) Kernel=15@(28, 1)	(-1, 15, 28, 178)	(-1, 15, 1, 89)
		Conv2d, LK-ReLU, AP((1,2)) Kernel=15@(14, 1), Stride=(14, 1)	(-1, 15, 28, 178)	(-1, 15, 2, 89)
		Concatenate, BN		(-1, 15, 3, 89)
Block3		Conv2d, LK-ReLU, AP((1,4)), BN, GAP Kernel=15@(3, 1)	(-1, 15, 3, 89)	(-1, 15, 1)
		Flatten	(-1, 15, 1)	(-1, 15,)
Fully connected layers		Linear(32), ReLU	(-1, 15,)	(-1, 32,)
		dropout(0.5)	(-1, 32,)	(-1, 32,)
		Linear(2)	(-1, 32,)	(-1, 2,)
		softmax	(-1, 2,)	(-1, 2,)

LK-ReLU is the Leaky-ReLU activation function. AP is the average pooling operation. BN stands for batch normalization. GAP is the global average pooling. '-1' in the tensor size stands for the number of samples within one mini-batch. The strides of CNNs are (1, 1) if not specified, and the one for pooling layers is the same as the pooling step.

CPU

CPU	RAM
305.00%	6.2 GiB
	Max 29GiB

```
Testing DataLoader 0: 100%

[2024-08-08 07:24:07] INFO (torcheeg/MainThread)
[Test] test_loss: 0.481 test_accuracy: 0.772
```

Test metric	DataLoader 0
test_accuracy	0.7716803550720215
test_loss	0.4811123013496399

```
Fold 4 test accuracy: 0.7717
```

CONCLUSION:

- ✓ Implemented dataset's supervised machine learning classification models.
- ✓ Emotion recognition is most suitable using deep learning techniques especially using TSCEPTION. (0.77%~0.78%)
- ✓ Upgraded upon original dataset's accuracy measures other evaluation metrics such as F1-score.

FAQ:

- ✓ Why did you use DREAMER dataset for emotion recognition?
  - ✓ SEED dataset comprises EEG recordings from 15 participants who each viewed 15 film clips, focusing on three emotional states: positive, neutral, and negative.
  - ✓ DEAP, which includes EEG and peripheral physiological signals from 32 participants who watched 40 music videos. (VAD Scale)
  - ✓ While DREAMER dataset contains both EEG and ECG data from 23 participants exposed to 18 film clips. (VAD Scale)

FAQ:

- ✓ Why did you use LSTM and CNN?
  - ✓ Both types are designed to process complex data.
  - ✓ LSTMs are particularly effective for learning from and making predictions based on time-series data, capturing long-term dependencies and non-linear features in sequential data.
  - ✓ While CNNs are proficient in detecting spatial patterns and features, making them useful for processing the multi-dimensional nature of physiological signals.

THANK YOU