Exploring Contribution of Individual Bio-Signals in Multi-Modal Emotion Recognition and Fusion Strategies using End-to-End Deep Learning

Master Thesis

for obtaining the academic degree:
Master of Science

presented by:

Ashutosh Singh

born on 14th August 1995 in Uttar Pradesh, INDIA

Matriculation Number: 22855339

Supervisor: PD Dr.-Ing. Thomas Wittenberg
Supervisor from Fruanhofer Institute: Dr. Dr. Theresa Götz

Start Date: 01.05.2022
Submission Date: 01.11.2022

Educational Institute: Friedrich-Alexander-Universität Erlangen-Nürnberg
Chair: Department Informatik
Lehrstuhl 9 - Graphische Datenverarbeitung
Friedrich-Alexander-Universität Erlangen-Nürnberg
Course: Computaional Engineering - Information Technology, Digital Signal Processing

Research Institute: Fraunhofer-Institut für Integrierte Schaltungen IIS

Theme: Exploring Contribution of Individual Bio-Signals in Multi-Modal Emotion Recognition and Fusion Strategies using End-to-End Deep Learning

Ashutosh Singh
Hasslebrookestraße 138
22089 Hamburg

E-Mail: ashutosh.singh@iis.fraunhofer.de
ashutosh.singh.de@gmail.com

Acknowledgements

I would like to thank many people who helped through suggestions and comments about the work presented. I specially appreciate and time and suggestions given by Dr. Thomas Wittenberg during the course of 6 months al the Multimodal Fusion Team at Fraunhofer IIS, Dr. Theresa Götz, Dr. Shrutika Sawant, Andreas Folyton, Jaspar Pahl, Stephan Göb and Jessica Deuschel.

This work was supported by the Bavarian Ministry of Economic Affairs, Regional Development and Energy through the Center for Analytics–Data–Applications (ADA-Center) within the framework of "BAYERN DIGITAL II"(20-3410-2-9-8).

Abstract

Human emotion recognition consists of large number of applications ranging from monitoring cognitive load to a more efficient Human-Computer Interaction. Facial expressions and bio-signals like electrocardiogram(ECG), electroencephalogram (EEG) play an important in recognizing emotions with high degree of certainty. In this work we will explore end-to-end deep learning based multi-modal fusion approaches for classifying(predicting) the emotional states ‘arousal‘ and ‘valence‘[1]. Instead of combining all the available modalities to improve performance of emotion recognition system, at once we will study the contribution of each of them in classifying the emotion. This analysis will help in understanding contribution of each modality and capturing the complementary information among different modalities.

We hypothesise that studying this individual contribution or discriminatory power might help us in developing more robust fusion strategies along with the added benefit of not having to use all the signals; which is very crucial keeping a real world scenario in mind where most of the modalities might not be available. We evaluate the performance of the proposed fusion strategies on publicly available MAHNOB-HCI[1] dataset. We first individually validate the performance obtained by each of modalities (or bio-signals). We then evaluate the performance of end-to-end deep learning based strategies by combining some as well as all of bio-signals. Furthermore, we pretrain the bio-signal feature extractor using an autoencoder in an unsupervised manner. This allows to use more training data mitigating the problem of small size of the dataset.

Contents

Nomenclature 7

1 Introduction 10
- 1.1 Emotion Recognition 11
- 1.2 Deep Learning 12
- 1.3 Unsupervised Learning 12
- 1.4 Autoencoders 12
- 1.5 Variational Autoencoders 13
- 1.6 LOSO 14
- 1.7 Loss Functions 14
  - 1.7.1 Mean Squared Error 14
  - 1.7.2 Cross-Entropy Loss 14
  - 1.7.3 KL Divergence 14
- 1.8 Performance Metrics 14
  - 1.8.1 Macro Classification Accuracy 14
  - 1.8.2 Macro F1-Score 15
  - 1.8.3 ROC-AUC 15
- 1.9 Datasets 15
  - 1.9.1 Dataset Summary 15
  - 1.9.2 Modalities 16
  - 1.9.3 Stimuli 17
- 1.10 Data Fusion 18
  - 1.10.1 Late Fusion 18
  - 1.10.2 Early Fusion 19
  - 1.10.3 Intermediate Fusion 19

2 Tools and resources 20
- 2.1 Neural Networks 20
  - 2.1.1 Fully Connected Networks 20
  - 2.1.2 Convolutional Neural Networks 20
  - 2.1.3 Recurrent Neural Networks 21
  - 2.1.4 Wavelet Decomposition Networks 22
  - 2.1.5 ConvLSTM 23
  - 2.1.6 WaveBlock-ConvLSTM 23
- 2.2 Software and Libraries 23

3 Data Preparation and Problem Statement 25
- 3.1 Data Preparation 25
  - 3.1.1 Final datasets 28
  - 3.1.2 Min-Max Normalization 28

3.2 Problem Formulation 29
  3.2.1 Unimodal Classification 29
  3.2.2 Decision Fusion(Late Fusion) 29
  3.2.3 Intermediate Fusion 29

4 Classification Algorithms and Architectures 30
4.1 Base Modules 30
4.2 Experimental Settings and Evaluation Metrics 32
4.3 Unimodal Classification 32
  4.3.1 Architectures 32
4.4 Decision Fusion 36
4.5 Intermediate Fusion Pipeline 37
  4.5.1 Architectures 38
4.6 Results 41

5 Pretraining of the Bio-Signal Feature Extractor 42
5.1 Data 42
5.2 Problem Formulation 43
5.3 Model 43
5.4 Training and Evaluation 43
5.5 Results 43

6 Analysis 45
6.1 Uncertainty Analysis 45
  6.1.1 Samples from extreme ends 47
  6.1.2 Uncertainty after fusion 49
6.2 Baseline vs Stimulus Periods 50
  6.2.1 Raw Signals 51
  6.2.2 Hand-Crafted Features 53
6.3 Latent Features Analysis 55
  6.3.1 Analyses of Latent Subgroups 55

7 Discussion 59

8 Future Works 61

9 Conclusion 62

10 Literature Research 63
10.1 Deep Learning based Fusion Methods 63
10.2 Machine Learning based Fusion Methods 64
10.3 Unsupervised Learning in context of physiological signals 65
10.4 Data 65

Appendix 71

Nomenclature

Vector and Scalar Notation

$\mathbf{A}$	Matrix
$\mathbf{a}$	Vector
a	Scalar

Symbols

$\mathcal{D}$	Dataset
$\mathcal{C}$	Set of classes
$\mathcal{M}$	Set of modalities
$\mathbf{W}$	Weight Matrices for Neural Nets
$\mathcal{S}$	Segment from a trial
r	Self-Rating for trial(and all its segments)
y	Binary Labels or trial(and all its segments)
$\hat{\mathbf{y}}$	Predicted label for a segment $\mathcal{S}$
m	Any modality from $\mathcal{M}$
l	Length of the segment vector
$\mathcal{T}$	Set of all subjects
$f(.;\theta)$ and $g(.;\phi)$	Parameterized functions
$\mathbf{l}$	Low-Pass Filter
$\mathbf{h}$	High-Pass Filter

Subscripts

i	Used to indicate i^{th} in a set. For example $\mathcal{S}_i$ is i^{th} segment in a dataset.
m	Used to indicate use of m^{th} modality
t	Given time-step in a sequence
n	Last time-step in a sequence

Acronyms

LOSO	Leave One Subject Out
ECG	Electrocardiogram
GSR	Galvanic Skin Response
EDA	Electrodermal Activity
EEG	Electroencephalogram
Resp	Respiration Amplitude
Temp	Temperature(Skin)
FC	Fully Connected
FCN	Fully Connected Network
CNN	Convolutional Neural Network
LSTM	Long Short-Term Memory
RNN	Recurrent Neural Network
LA/HA	Low/High Arousal
LV/HV	Low/High Valence
LPF/HPF	Low/High Pass Filter
MMD	Maximum-Mean Discrepancy
KLD	Kullback-Liebler Divergence
MSE	Mean-Squared Error
CE	Cross Entropy
VAE/AE	Variational Autoencoder/Autoencoder
TP/FP	True/False Positive
TN/FN	True/False Negative

1 Introduction

Affective computing is the study and development of systems and devices that can recognize, interpret, process, and simulate human affects. It is an interdisciplinary field spanning computer science, psychology, and cognitive science[2]. One of the main purposes of affective computing is to enable machines to better understand human emotional state and accordingly assist them in different situations[3]

Emotion detection research is becoming very popular due to wide range of applications. Human emotions are very complex and are reflected completely/partially in multiple types of signals like facial expressions, heart rate, body temperature, brain function etc.

Current datasets available for emotion detection generally contain these signals in form of multiple modalities like Facial Video/Audio, ECG, Galvanic Skin Response(GSR), Respiration Amplitude, Eye Gaze Tracking data and Electroencephalogram(EEG). While each of these modalities plays an important role in human emotion detection some of the signals are more difficult and expensive to record than others.

In this work we focus on signals that are less expensive to record and study how each signal performs in recognizing the emotion on its own and together with other signals.

Existing works focus on both visual data i.e facial expressions' videos and biological signals like ECG, EEG etc to detect emotions.

Facial expressions effectively reflect emotion and are easy to capture with one or more video camera. These works have successfully exploited both hand-crafted features and deep learning based methods to detect emotions. Hand-crafted features include facial landmark detection[4], Histogram of Gradients(HoG)[5], spatiotemporal descriptors[6] etc. and deep learning methods include various architectures for processing series of face images, for example ConvLSTM[7], 3D-CNN[8] and CNN-RNN[9] based approaches to model the data.

Biological signals which include Electroencephalogram (EEG), Electrocardiogram (ECG), and Galvanic Skin Response (GSR) etc. are also used in recent works to interpret human emotions. The motivation of using these bio signals comes from the fact that emotions can be concealed and may not be reflected as evidently in facial expressions as in these signals since they are more spontaneous and hard to control for humans. Emotion detection using bio-signals has been explored both as classification of handcrafted and statistical features and also deep learning methods for example [10] to extract features.

In addition to emotion detection using individual bio-signals, fusion of multiple modalities to increase the accuracy of the system has also been explored. There have been a number of studies both for decision level fusion and feature level fusion of bio-signals. Related works include both classical ML-Methods and DL based methods for fusing multiple modalities together. [11] use discriminative handcrafted features and feature level fusion with classifier such as SVM and KNN etc. [12] fuse EEG with face video modality for feature level fusion and [13] use feature fusion and decision level fusion for video modality and bio-signals to make final decision.

The goal of this work is to use end-to-end deep learning methods for robust fusion strategies. We want to exploit local spatial structures in bio-signals and temporal nature of the signals.

We hypothesize that emotion is localized within certain regions from entire experiments and we want to incorporate this when using multi-modal fusion.

1.1 Emotion Recognition

Emotion Recognition is the process by which we can distinguish one emotion from other. Emotion recognition is an active research topic in Affective Computing.
There are multiple ways to formulate the problem of emotion classification such as classifying between discrete emotions like *joy*, *anger*, *sadness*, *disgust* etc. or using a dimensional model for emotions and associating each emotion to a point in this space.
There are multiple dimensional models for emotions and the models themselves are out of scope for this study, however most of these models use *valence* and *arousal* dimensions and all emotions can be represented in this 2-Dimensional Space as can be seen in figure-3.3.

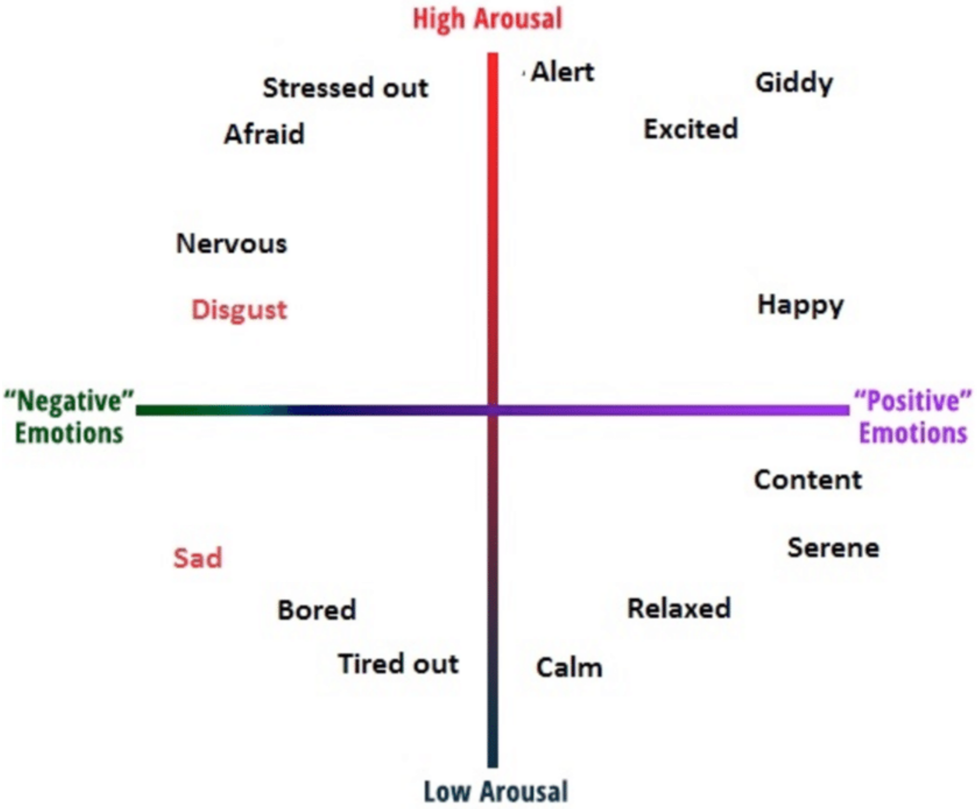


Figure 1.1: Valence(negative/positive) and Arousal(low/high) as the two dimensions of emotions[14]

Valence and Arousal values generally range from 1 to 9, however some datasets record valence and arousal on the scale of 1 to 5, for example DREAMER dataset[15].

1.2 Deep Learning

Deep Learning although needs no introduction, we very briefly introduce it using the descriptions provided by Yarin Gal[16]. They describe deep learning by extending on the *linear basis function regression* which is common choice over *linear regression* if the relationship between independent variables $\mathbf{x}$ and dependent variable $\mathbf{y}$ is non-linear. Linear basis function regression is essentially linear regression over a feature-vector of scalar-valued *non-linear transformations* of $\mathbf{x}$

$$\Phi(\mathbf{x}) = \{\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), .., \phi_n(\mathbf{x})\} \quad \ni \phi_t(\mathbf{x}) \in \mathbb{R} \quad \forall \phi_t$$

To maintain the assumption of independence of features in $\mathbf{x}$(for linear regression) the basis functions ϕ_t are assumed to be orthogonal and fixed.
If these conditions are relaxed we can further deepen the idea of basis functions by making them *parameterized* as well $\phi_t^{\mathbf{w}_t,\mathbf{b}_t}$ where instead of applying ϕ_t

$$\phi_t^{\mathbf{w}_t,\mathbf{b}_t} = \phi_t(\mathbf{w}_k.\mathbf{x} + \mathbf{b}_t)$$

Yarin then describes fully-connected networks, the most basic idea in deep learning as a *hierarchy* of these parameterized basis functions where each feature vector $\phi(\mathbf{x})$ is called a *layer*.

1.3 Unsupervised Learning

Unsupervised learning is a type of algorithm that learns patterns from unlabelled data. The hope is that through mimicry, which is an important mode of learning in people, the machine is forced to build a compact internal representation of its world and then generate imaginative content from it.
In contrast to supervised learning where data is tagged by an expert, e.g. as a ”ball” or ”fish”, unsupervised methods exhibit self-organization that captures patterns as probability densities or a combination of neural feature preferences[17]

1.4 Autoencoders

Autoencoders are an Unsupervised Representation Learning technique that learn representations from unlabelled data.
An Autoencoder has two components an Encoder and a Decoder. The encoder is a neural network that encodes the input x to a dense feature vector z. The decoder is again a neural network which uses these features to reconstruct the input as $\hat{x}$.
The whole system can be trained end-to-end by using Backpropagation to minimize the L2-Distance between input x and reconstructed version $\hat{x}$.

$$Loss = ||x - \hat{x}||_2^2 \tag{1.1}$$

Autoencoders have been shown to learn very rich feature representations from unlabelled data. These features can be used in a supervised problem such as classification by throwing away the decoder and attaching a classification head on the encoder

1.5 Variational Autoencoders

Variational Autoencoder can be seen as a generative autoencoder. It has more efficient probabilistic structure that allows for its use as a Generative model.
Variational Autoencoder tries to approximate the data generation process $p(x)$ by defining a latent space with a known prior usually Gaussian, $p(z)$. $p(x)$ can be described as

$$p_\theta(x) = \int p_\theta(x, z) = \int p_\theta(x|z) p_\theta(z) dz \tag{1.2}$$

The distribution $p_\theta(x|z)$ is modelled by a neural network, the decoder network. $p_\theta(z)$ is a known distribution, we assume a diagonal Gaussian Distribution. However this integral is intractable since it is over all z
The likelihood to be maximized can be written using Bayes' rule.

$$p_\theta(x) = \frac{p_\theta(x|z) p_\theta(z)}{p_\theta(z|x)} \tag{1.3}$$

The denominator $p_\theta(z|x)$ is again intractable and another neural network is used to approximate this as $q_\phi(z|x)$, the encoder network.

$$p_\theta(x) \approx \frac{p_\theta(x|z) p_\theta(z)}{q_\phi(z|x)} \tag{1.4}$$

In a Variational Autoencoder the encoder learns a distribution over the latent variables z, $q_\phi(z|x)$ and the decoder samples from the prior $p_\theta(z)$ and learns a distribution over x, $p_\theta(x|z)$.
Since sampling is not a differentiable operation, variational autoencoders employ the Reparametrizarion trick. The encoder outputs two numbers μ and σ and then z is calculated as:

$$z = \mu + \sigma \epsilon, \;\; \epsilon \sim \mathcal{N}(0, 1) \tag{1.5}$$

where:
$\mathcal{N}(0, 1) :=$ Standard Normal Distribution

The Reparameterization trick enables the Variational Autoencoders to be trained end-to-end by maximizing the log likelihood function which can be proved to come out as

$$\log p_\theta(x|z) = \mathbb{E}_{z \sim q_\phi(z|x)}[\log p_\theta(x|z)] - \mathcal{D}_{KL}(q_\phi(z|x), p(z)) + \mathcal{D}_{KL}(q_\phi(z|x), p_\theta(z|x)) \tag{1.6}$$

The last term in this equation i.e. the KL-Divergence between encoder and posterior of decoder is again intractable since we can't compute the posterior distribution. Since KL-Divergence is a number greater than zero this term can be dropped to give a lower bound on the log-likelihood.

$$\log p_\theta(x|z) \geq \mathbb{E}_{z \sim q_\phi(z|x)}[\log p_\theta(x|z)] - \mathcal{D}_{KL}(q_\phi(z|x), p(z)) \tag{1.7}$$

Now the encoder and decoder can be jointly trained to maximize this Variational Lower Bound or Evidence Lower Bound(ELBO).

1.6 LOSO

Leave One Subject Out(LOSO) is a common testing strategy for models developed for physiological datasets. The idea behind LOSO is to keep one subject completely out of training data and use it for testing. The main goal of this method is to prevent subject level bias in the modelling process.
Despite multiple advantages of this method its not without bias[18].

1.7 Loss Functions

1.7.1 Mean Squared Error

MSE is a commonly used loss function in regression tasks. It is the average of squared difference between target and model's predictions over the dataset. MSE is always positive due to the squaring operation involved.

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [Y(i,j) - \hat{Y}(i,j)]^2 \tag{1.8}$$

1.7.2 Cross-Entropy Loss

Cross-Entropy loss is used to measure the performance of a classifier whose output is a probability distribution. This loss has its roots from Information Theory.

$$CE(Y, \hat{Y}) = \sum_{c \in \mathcal{C}} -Y_c \log \hat{Y}_c \tag{1.9}$$

1.7.3 KL Divergence

KL Divergence is used to measure the divergence of one probability distribution from the other. It is a not a proper distance metric. It has many applications in Unsupervised Learning where we try to model the distribution directly.

$$\mathcal{D}_{KL}(\mathcal{P} \| \mathcal{Q}) = \sum_{x \sim \mathcal{X}} \mathcal{P}(x) \log \frac{\mathcal{P}(x)}{\mathcal{Q}(x)} \tag{1.10}$$

1.8 Performance Metrics

1.8.1 Macro Classification Accuracy

Macro accuracy is used to measure the performance of a classifier. This measure is especially preferred over its close relative Micro Accuracy(or Accuracy) if there is an imbalance in test dataset.Macro accuracy is calculate as average of accuracy computed for each class separately. It ranges from 0 to 1.

$$Macro - Accuracy = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} Accuracy_c$$

where $Accuracy_c$ is accuracy for class c.

1.8.2 Macro F1-Score

Macro F1-Score is the average of F1-Score calculated for each class in the dataset separately. F1-Score ranges from 0 to 1.

$$Macro - F1 = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} F1_c$$

where $F1_c$ is F1-Score for class c.

1.8.3 ROC-AUC

Receiver Operating Characteristic curve is a graph showing performance of a classification model at different *decision thresholds*. The plot consists of TPR(True Positive Rate) and FPR(False Positive Rate)

$$TPR = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN}$$

AUC(Area Under Curve) of the ROC curve provides aggregate classification performance across all descision thresholds. AUC can be interpreted as the probability that the model ranks a random positive example higher than random negative example. AUC ranges from 0 to 1.

1.9 Datasets

We use MAHNOB-HCI[1] dataset for this study.

1.9.1 Dataset Summary

Participants and Modalities	
Number of Participants	27, 11 Male and 16 Female
Recorded Signals	32-Channel EEG, Peripheral Physiological Signals(256Hz), Face and Body Video using 6 Cameras(60fps), Eye Gaze Data(60Hz), Audio(44.1kHz)
Emotional Responses to videos(Experiment 1)	
Number of Videos	20
Selection Method	Subset of online Annotated Videos
Self Report	Emotional Keyword arousal, valence, domninance, predictability
Rating Values	Discrete Scale 1-9
Implicit Tagging(Experiment 2)	
Nr. of Videos and Images	14 Videos, 28 Images
Dataset	Pictures from flickr and short videos
Self Report	Agreement with displayed tags

Table 1.1: Summary table for MAHNOB-HCI dataset. Source [1]

1.9.2 Modalities

Multiple bio-signals are collected in the MAHNOB-HCI dataset. Below is a list of signals that will be used for valence-arousal detection in this study. We will refer to these signals by the acronyms of their names as specified in the first column.

Acronym	Name
ECG	Electrocardiogram
GSR(EDA)	Galvanic Skin Response (Electrodermal Activity)
Resp	Respiration Amplitude
Temp	Body Temperature
EEG	Electroencephalogram

Table 1.2: List of physiological signals used for this study

ECG

ECG signals contain the information about rhythm and electrical activity of heart.
For ECG data collection three electrodes were attached to participants' body. Two electrodes were placed on chest's upper right and left corners below the clavicle bones and the third electrode was placed on the abdomen below the last rib.

GSR

Galvanic Skin Response(Electrodermal Activity) is the measure of changes in 'sweat gland' activity which is an indicator of the intensity of our emotional state.
For GSR measurement in MAHNOB-HCI data collection, 2 electrodes are placed on the distal phalanges of the middle and index fingers and a negligible amount of current is passed through the body. Perspiration causes changes in resistance to this current, which is then measured through the electrodes as a 1-channel signal.[1]
The perspiration changes are usually caused by emotions such as stress or surprise. Relationship between GSR and emotional state has been shown in research [19] [20]

Resp

Respiration Amplitude(Resp) is measured as the physical change of the thoracic expansion with belt around participant's abdomen. Resp is closely linked to heart activity and emotional state.[21].
MAHNOB-HCI dataset has a 1-channel RA signal which is measured as described above.

Temp

Temp(in context of this study) is a measure temperature of the outermost layer of skin. The normal range of skin temperature for human varies between 33.5 and 36.9 °C. Relationship between skin temperature and emotional state has been studied in research works like[22] [23] [24]
The temperature was recorded as a 1-channel signal by placing a sensor on participants' little finger.

1.9.3 Stimuli

MAHNOB-HCI dataset employs videos(and images) as stimuli to elicit emotional response from the participants.The emotion elicitation experiment contains 20 video clips which were selected from multiple commercially produced movies. The authors did a preliminary online study to get emotion tags corresponding to each video clip where each clip received at least 10 annotations from over 50 participants.

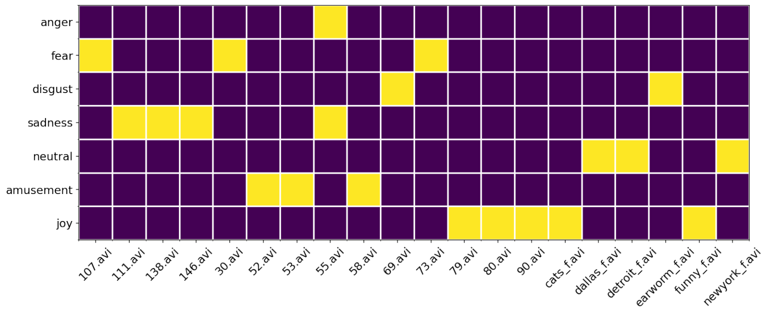


Figure 1.2: Emotion keywords associated to video clips used from preliminary study

These video clips were then shown to 30 participants where after watching the video participants were asked to give a self-rating for valence and arousal on a 1-9 scale.Since the participants provide different ratings to each video clip, we wanted to analyze if there is any agreement between these ratings. The visualizations presented below confirm that in most cases majority of participants agree on a given rating for both Arousal and Valence; however there are quite significant differences as well. For example in case of valence we see much more significant spread of for some clips like **earworm_f.avi**. Arousal ratings are much more spread out than valence, even for high arousal stimuli like joy(**79.avi**, **80.avi** and **90.avi**)

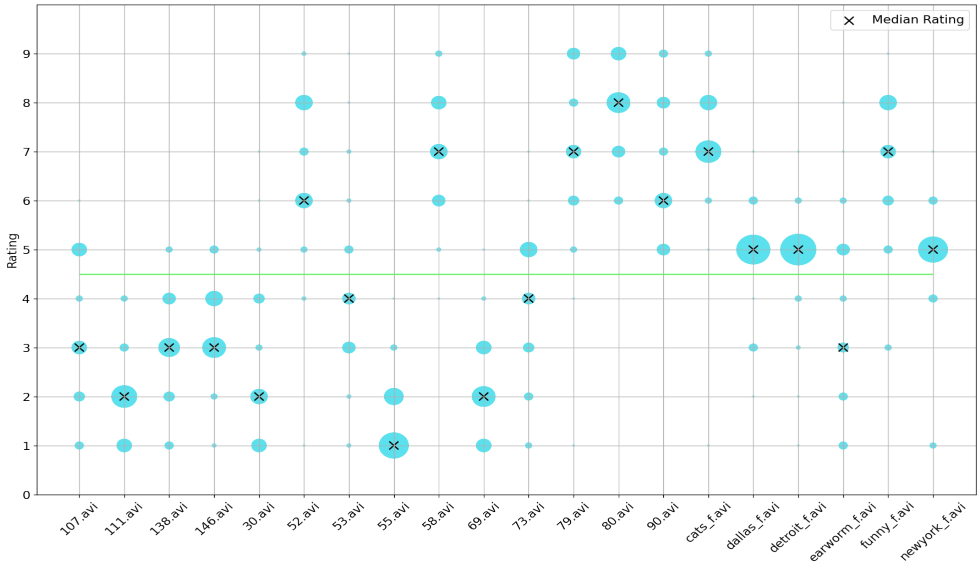


Figure 1.3: Valence ratings for video clips. The size of the dots is proportional to number votes for given rating.

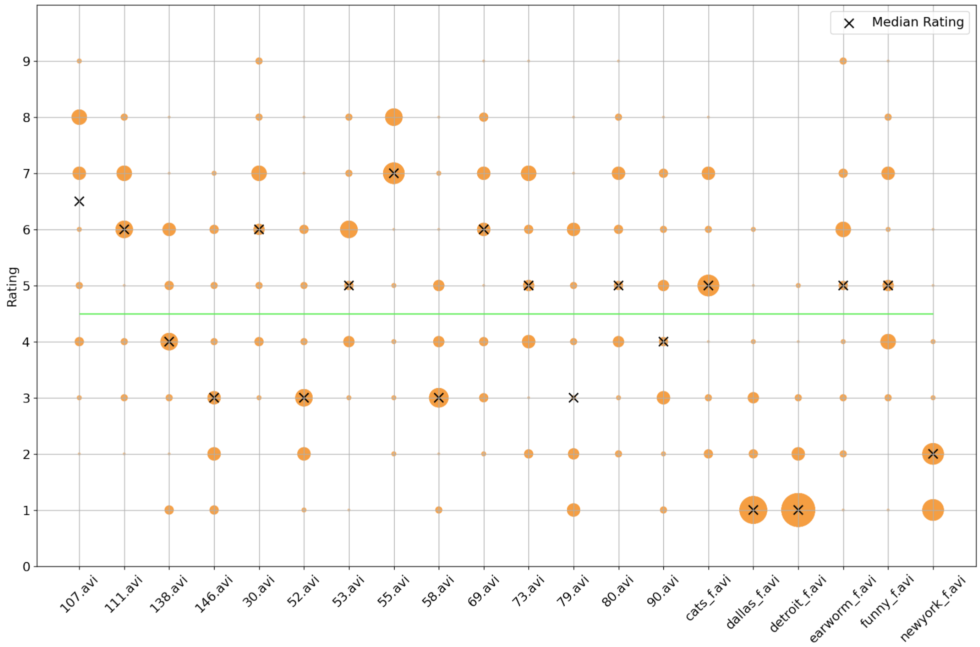


Figure 1.4: Arousal ratings for video clips. The size of the dots is proportional to number votes for given rating.

1.10 Data Fusion

Data fusion is the process combining data from multiple modalities with the goal of getting complimentary information from each modality to get more accurate representations, which are better than those provided by individual modalities for machine learning tasks like classification/regression.
Fusion strategies are mainly grouped into three categories namely early fusion, intermediate fusion and late fusion.

1.10.1 Late Fusion

Late fusion also known as *decision fusion*, is the process of combining decision made by the machine/deep learning model. For example in case of classification the probabilities of different classes may be averaged and normalized over all modalities, or some boosting techniques.

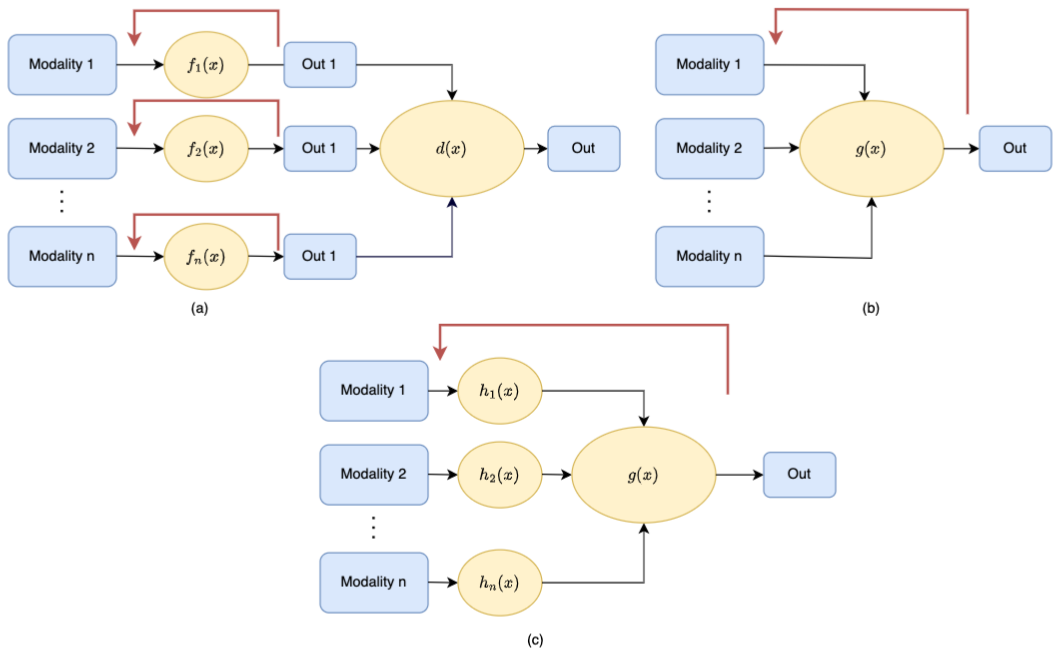


Figure 1.5: Illustrations for different fusion strategies. (a) Late Fusion, (b) Early Fusion, (c) Intermediate Fusion. Functions represent parameterized models, red arrow represents error backpropagation during training.

1.10.2 Early Fusion

Early fusion refers to the process of joining/combining multiple modalities together before feeding them to the machine/deep learning model. The modalities maybe combined in multiple ways for example, channel-wise concatenation of PSD based images.

1.10.3 Intermediate Fusion

It is the process of combining intermediate representations from different modalities. These intermediate representations are learnt separately for every modality before the fusion step.
A simple example is having different encoder for each modality and then fusing the representations from these encoders via some interaction. In this work we focus on intermediate fusion.

2 Tools and resources

In this chapter we introduce all the tools and resources that were used during the course of this study. We primarily introduce the types of Neural Networks we will be using along with any other software libraries.

2.1 Neural Networks

2.1.1 Fully Connected Networks

Fully-Connected Neural Networks also known as *Feed Forward Neural Networks*[25] are the most basic type of neural networks. They are a hierarchical structure of affine transformations and element-wise non-linearity[16].

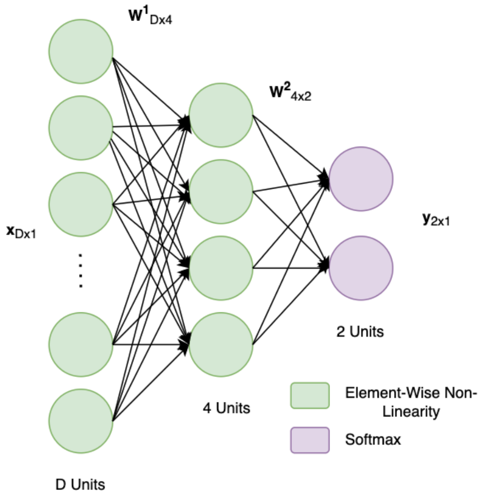


Figure 2.1: Illustration of a 2 layer FCN. Input $\mathbf{x}$ goes through an affine transformation $\mathbf{W}^T.\mathbf{x}+\mathbf{b}$. $\mathbf{W}$ and $\mathbf{b}$ are the *weights(linear map*) and *bias(translation)* for FC layers. The last layer has a softmax non-linearity for classification problems.

2.1.2 Convolutional Neural Networks

CNNs([25], [26]) are a type of neural networks that use convolution operation along with pooling operation to process spatial data.
Primarily developed and used for *image processing* because of their strong inductive bias of rotational and transational invariance; properties very much desirable for image processing. They are also capable of handling scale invariance, another important property by repetitive application of convolution and pooling operations multiple times.
Application of CNNs is not limited to image processing. They have been known to process sequence data like time series and text processing as well. In this work we use 1D-CNNs to

extract features and also in places to reduce overall temporal length of the time series signals like ECG.

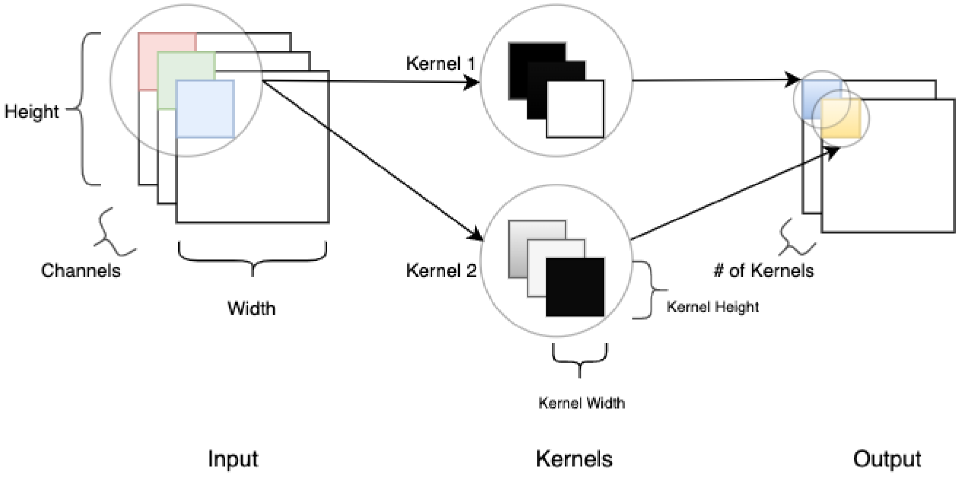


Figure 2.2: Illustration of one CNN layer. Input of given height and weight with given number of channels and two kernels. Each patch in input(like the one shown on top-left) is convolved with each kernel. First kernel only selects blue to generate blue pixel in the output while second kernel ignores blue to output a yellow pixel[16]

2.1.3 Recurrent Neural Networks

RNNs([25], [27]) are a special type of neural networks that deal with sequential data. They are recursive in nature i.e. model parameters are shared by each timestep.They leverage the sequential dependence by using the outputs from previous timestep in the current timestep. In this work we mainly use LSTM(Long Short Term Memory)[28], a variant or extention of the RNN. Th main advantage of LSTM over an RNN lies in its ability to deal with the problem of *vanishing gradient descent*. LSTM uses an extra gate to forget information from the past.

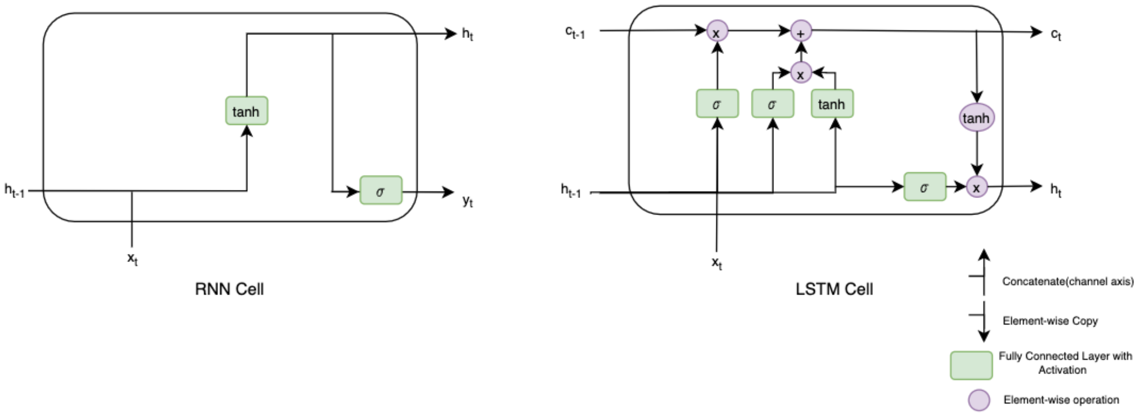


Figure 2.3: Illustration of RNN Cell on left and an LSTM Cell on the right.

2.1.4 Wavelet Decomposition Networks

mWDN[29] or WDN are special types of neural networks based on the idea of Wavelet Decomposition. *Multi-level Wavelet Decomposition* splits a time series signal into low- and high-frequency sub-series. This decomposition into sub-series in done repetitively to get multi-level time-frequency features.
The original series is passed through a Low Pass Filter and a corresponding High Pass Filter at each level to get the sub-series. These filters are fixed(assumption for simplicity, even if they are fixed; they are not learnt through feature extraction).
mWDNs overcome this limitation of fixed parameters by learning the low and high pass filters at each level for sub-series formation.
The idea is really simple and intuitive, given a time series $\mathbf{x} = \{x_1, x_2, ..., x_T\}$, a low pass filter $\mathbf{l} = \{l_1, l_2, ..., l_K\}$ and a high pass filter $\mathbf{h} = \{h_1, h_2, ..., h_K\}$ wavelet decomposition breaks $\mathbf{x}$ into low and high frequency sub-series $\mathbf{x}^l(i)$ and $\mathbf{x}^h(i)$ at the i^{th}-level(2.5) through convolutions Each level of mWDN is denoted as *WaveBlock*.

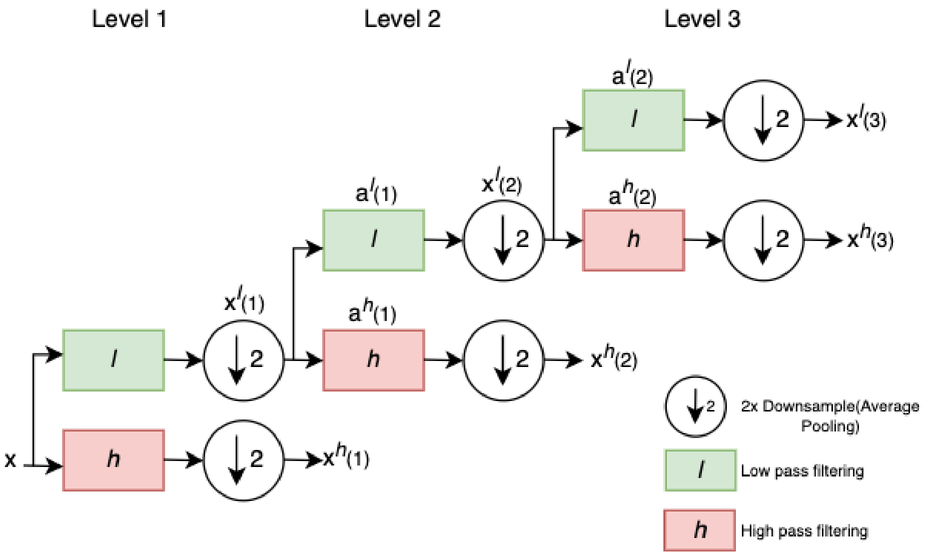


Figure 2.4: mWDN Framework for 3 decomposition levels.[29]

$$a^l(i+1) = \Sigma_{k=1}^{K} x_{n+k-1}.l_k$$
$$a^h(i+1) = \Sigma_{k=1}^{K} x_{n+k-1}.h_k$$

mWDN replaces these convolutions by fully connected layers at each decomposition level.The weight matrices $\mathbf{W}^l(i)$ and $\mathbf{W}^h(i)$ are initialized as circulant matrices with coefficients of low and high pass filters.

$$\mathbf{a}^l(i+1) = \sigma(\mathbf{W}^l(i+1).\mathbf{x}(i)^l + \mathbf{b}^l(i+1))$$
$$\mathbf{a}^h(i+1) = \sigma(\mathbf{W}^h(i+1).\mathbf{x}(i)^h + \mathbf{b}^h(i+1))$$

where:

$$\mathbf{W}^{l}(i+1) = \begin{vmatrix} l_1 & l_2 & \dots & l_K & \epsilon & .. & \epsilon \\ \epsilon & l_1 & l_2 & \dots & l_K & \dots & \epsilon \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ \epsilon & \epsilon & l_1 & l_2 & \dots & l_K & \epsilon \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ \epsilon & \epsilon & \ddots & \epsilon & l_1 & l_2 & l_3 \end{vmatrix}$$

$\mathbf{W}^{h}(i+1)$ follows the same structure as $\mathbf{W}^{l}(i+1)$ with the difference that the coefficients come from the *high-pass* filter **h**.

2.1.5 ConvLSTM

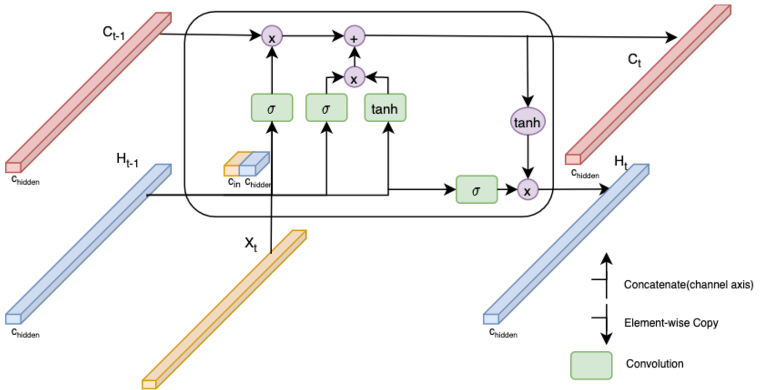


Figure 2.5: Cell structure of ConvLSTM and how the input is concatenated in the *channel* dimension

ConvLSTM[30] is type of neural network that treats data as a spatio-temporal sequence. Major difference between a vanilla LSTM Cell and ConvLSTM is how we model the gates inside the cell. In LSTM these gates are modelled by 4 FC layers; while for ConvLSTM they are modelled as Convolutional layers.

2.1.6 WaveBlock-ConvLSTM

We extend the ConvLSTM further to WaveBlock-ConvLSTM. We hypothesize that this network can also extract frequency-domain information at each time-step. We replace the convolutional layers inside the ConvLSTM cell with a WaveBlock from mWDN followed by a convolutional layer for each LSTM gate.

2.2 Software and Libraries

We use Pytorch[31] for developing the deep learning models used in this work. The train and test time metrics are measured by using functionalities provided by TorchMetrics[32] library which lets the user measure and update metrics after each batch is processed.

We heavily rely on Matplotlib[33] for all the charts and graphs presented, along with Numpy[34] and SciPy[35] for various scientific computations we do during the course of this study. We also use the implementations of multiple algorithms from the Scikit-Learn[36] library for analysis with TSNE being the major one.
We use Captum[37] for measuring gradient attributions since captum can be very easily integrated with any pytorch model.

3 Data Preparation and Problem Statement

Now that we have introduced the tools we use for processing the data, in this chapter we describe the process of data preparation for actual valence and arousal classification. Furthermore, the problem statement is also presented.
We start with raw data and then describe in detail the steps we take to convert it into classification dataset. Along the way we also introduce some details about raw data and self-rating system to better explain our method.
In the problem statement section we introduce the structure of classification problems that we deal with in this study. Since the study is about data-fusion(see section-1.10) we introduce a set of sub-problems.

3.1 Data Preparation

In MAHNOB-HCI dataset each instance where a participant watches a video and gives a rating is referred to as a *trial*. Since the *stimuli*(see section-1.9.3) are of different lengths each trial as well has a different length.
The total number of labelled trials are very less for MAHNOB-HCI dataset with only 547 trials available for classification. This number is too small for Neural Networks so we divide each trial into smaller fixed length segments which are the used for classification as shown in Figure-3.1. As we know each trial has corresponding ratings of valence or arousal which is inherited by each segment from that trial.

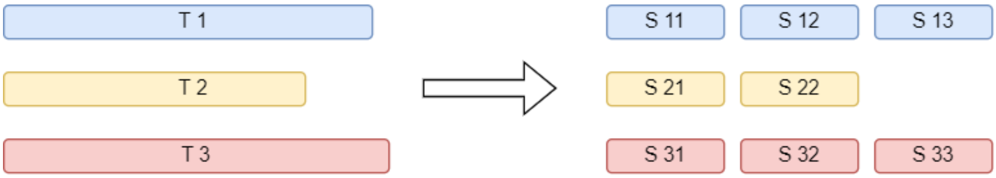


Figure 3.1: Illustration of Segmentation process. Example: Trial *T-1* to equal length segments *S-11*, *S-12*, *S-13*

After segmentation we have $\mathcal{S}_{ijm}$ segment, j^{th} segment from i^{th} trial and m^{th} modality.
Let's simplify the naming by removing the trial index to get segments $\mathcal{S}_{im}$ with shape $l_i \times c_m$ where l_i is the length of the segment vector, c_m is the channels in given modality.

Modality	c_m
ECG	3
GSR	1
Resp	1
Temp	1

Table 3.1: Channels for Bio Signals

Each segment has a valence/arousal self-rating associated to it $r_i \in \{1, 9\}$. Notice no subscript of m since this rating is same for all modalities. We transform this 9 point scale to a binary scale for simplification to get final labels $y_i \in \{0, 1\}$

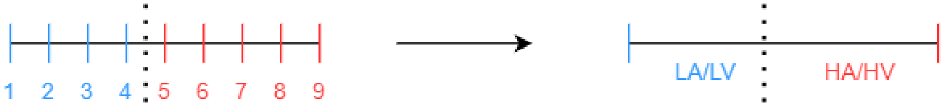


Figure 3.2: Transforming the nine point scale to a binary scale. 0 for LA/LV and 1 for HA/HV

We have selected a threshold of 4.5 for this transformation. We notice that this threshold gives the most balanced High vs Low class distribution.

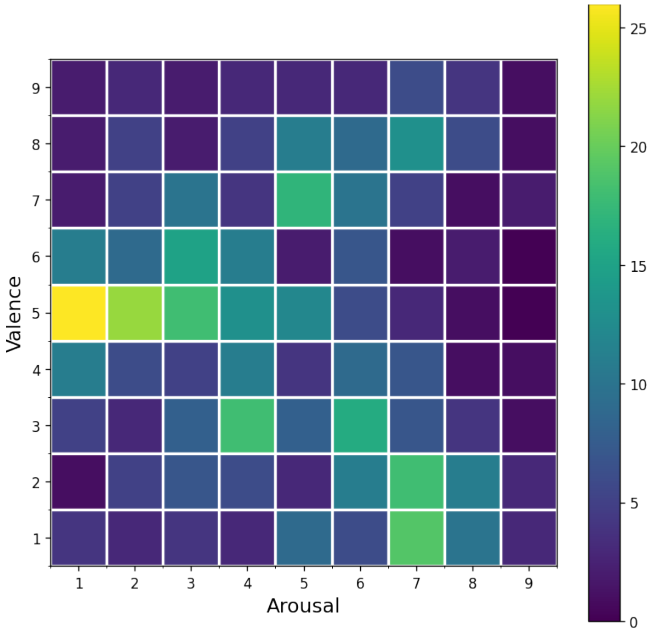


Figure 3.3: Variation of Valence with Arousal.

In addition to unbalanced classes, we point out(from Figure 3.3) that large number of samples have a rating of 5 which leads to large bias if 5 is treated as LA/LV.

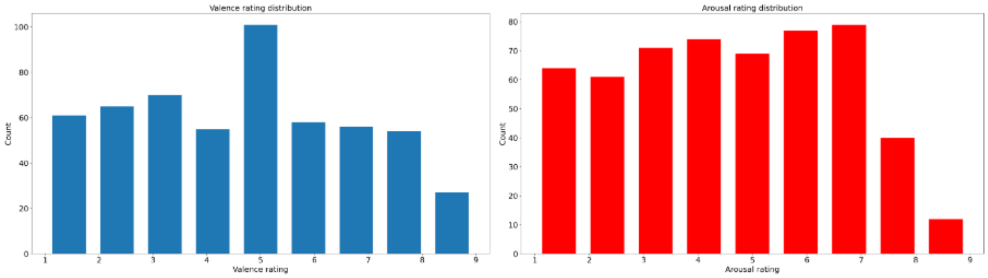


Figure 3.4: Rating distribution for Valence(left) and Arousal(right).

Further, there is trade-off between number of samples in high/low class given participant rating and preliminary emotion tag(see section 1.9.3); when transforming participant rating to binary scale using this *threshold*(=4.5).

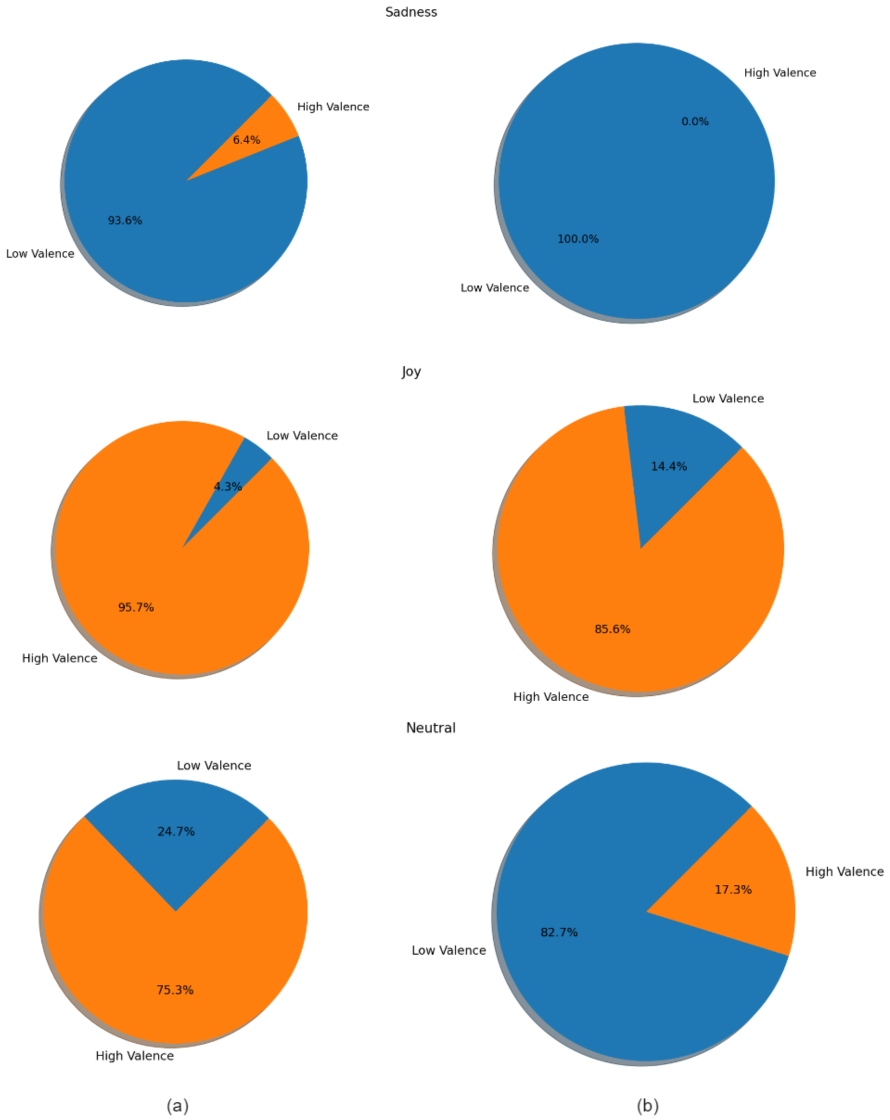


Figure 3.5: Distribution of High/Low Valence samples for preliminary tags of sadness(Low Valence), Joy(High Valence) and Neutral for (a) threshold = 4.5 and (b) threshold = 5.5

The figure above shows the class balance in data coming from media with preliminary emotion tags of Sadness, Joy and Neutral. We present these distributions for both thresholds of 4.5 and 5.5.
From Figure-3.5 we can see the trade-off in class distribution based on value of threshold, for example for preliminary tag *sadness* where for threshold value of 4.5(left column) we see some high valence samples but for threshold of 5.5 all the samples fall under low valence class(which is correct for sadness emotion); while for *joy* the case is opposite; i.e we lose samples from high valence class(correct class for joy).

After transforming the rating scale to binary and segmenting trials we have the classification dataset; a set of tuples for each modality:

$$(\mathcal{S}_{im}, y_i) \in \mathcal{D}_m \quad \forall m \in \mathcal{M}$$

where $\mathcal{M}$:= Set of all Modalities.

3.1.1 Final datasets

We create multiple versions of dataset by having different segment lengths and overlap values.
We perform *Segmentation* with two segment-lengths of 10s and 34s. 10s segments have an overlap of 5s while for 34s segments there is no such overlap. Furthermore, in 34s segments, we only use the last segment from each *trial*.

Segment Length(secs)	l_i	Sampling Rate	Overlap	Remark
34s	8704	256	0	Only the last segment is used
10s	2560	256	1280(5s)	-

Table 3.2: Details for 34s and 10s segments

where: l_i = Segment Length(secs) * Sampling Rate := Length of Segment Vector.

We assign names for datasets prepared as result of these segmentation lengths. These names are then used to refer the corresponding version in the rest of the studies.

Segment	Dataset Version Name
34s	MAHNOB-Data-V1
10s	MAHNOB-Data-V2

Table 3.3: Dataset names corresponding to each segment type

3.1.2 Min-Max Normalization

The bio-signals for different subjects have different range of values. The baseline values also increase or decrease based on position of sensor or belt(for Resp).
We apply min-max scaling to bring the scales for all subjects between 0 and 1.

$$x'_t = \frac{x_t - x_{t_{min}}}{x_{t_{max}} - x_{t_{min}}}, \quad t \in \mathcal{T} \tag{3.1}$$

where:
$\mathcal{T}$:= Set of all test subjects
$x_{t_{min}}$:= Min value of each modality for test subject t
$x_{t_{max}}$:= Max value of each modality for test subject t

3.2 Problem Formulation

Now that we have the classification dataset ready we describe the classification problem in detail.We start with a general classification problem statement.
Given a dataset of tuples $(\mathcal{S}_{im}, y_i)$ and we want to learn a parameterized model f_θ such that:

$$f_\theta(\mathcal{S}_{im};\theta) = \hat{\mathbf{y}}_i \quad \ni \operatorname{argmax} \hat{\mathbf{y}}_i \approx y_i \tag{3.2}$$

We treat HA vs LA(or HV vs LV) classification as a binary classification problem hence $\hat{y}_i \in [0,1]\times[0,1] \subset \mathbb{R}^2$.

We expand the idea of the general binary classification problem described above and formulate the problems of Unimodal Classification, Decision Fusion and then move to describe Intermediate Fusion.

3.2.1 Unimodal Classification

Unimodal Classification problem deals with using individual modalities for classification. We learn a set of m parameterized models one for each modality such that:

$$f_m(\mathcal{S}_{im};\theta_m) = \hat{\mathbf{y}}_{im} \quad \ni \operatorname{argmax} \hat{\mathbf{y}}_{im} \approx y_i \quad \forall m \in \mathcal{M} \tag{3.3}$$

3.2.2 Decision Fusion(Late Fusion)

Decision Fusion problem builds on top of unimodal classification and uses the results from individual modality classifiers. For Decision Fusion we use the set of outputs from unimodal classifiers $\{\hat{\mathbf{y}}_{im}\}$ to learn a **deterministic function d such that:

$$d(\{\hat{\mathbf{y}}_{im}\}) = \hat{\mathbf{y}}_i \quad \ni \operatorname{argmax} \hat{\mathbf{y}}_i \approx y_i \quad \forall m \in \mathbf{m} \subset \mathcal{M} \tag{3.4}$$

Where is **m** is the set of modalities being used for fusion.
**d need not be deterministic in general.

3.2.3 Intermediate Fusion

These fusion approaches have no relation to unimodal classification and uses multiple modalities to learn a parameterized model f end-to-end such that:

$$f(\{g_m(\mathcal{S}_{im};\theta_m)\};\phi) = \hat{\mathbf{y}}_i \quad \ni \operatorname{argmax} \hat{\mathbf{y}}_i \approx y_i \quad \forall m \in \mathbf{m} \subset \mathcal{M} \tag{3.5}$$

Where is **m** is the set of modalities being used for fusion.
Here the set of modality specific functions $\{g_m(.;\theta_m)\}$ and the final fusion function $f(.;\phi)$ are learnt together.Since we use end to end models so we simplify this equation to

$$f(\{S_{im}\};\phi) = \hat{\mathbf{y}}_i \quad \ni \operatorname{argmax} \hat{\mathbf{y}}_i \approx y_i \quad \forall m \in \mathbf{m} \subset \mathcal{M} \tag{3.6}$$

4 Classification Algorithms and Architectures

We have described 3 problems of Unimodal classification, Decision Fusion and Intermediate Fusion along with the datasets used for each of them in the previous chapter.
In this chapter we report the algorithms, observations and results for all three problems. We first describe some common components in the Base Modules Section(4.1). These components are used in different pipelines over the course of this study.
We also describe the experimental settings in Section-4.2. We use these settings for training all the models. We stick to these fixed settings so that we can better compare the results coming from different models.
After we describe the Base Modules and experimental settings we describe experimentation with unimodal classifiers for ECG, GSR, Resp and Temp. Our motivation for starting with single modalities was to analyze how each individual modality performs and if there is some correlation amongst predictions made by these individual signals for classification of Valence and Arousal.
In decision fusion we briefly mention the motivation behind this method and report the results and the algorithm used. We use a simple Decision Fusion algorithm to serve as one of the baselines for all fusion results. Finally we report on intermediate fusion including shortcomings of decision fusion, general motivation behind intermediate fusion and experimental results.

The results of individual classifiers are listed in the respective sections and the final comparison is done in the Results(4.6) section.

4.1 Base Modules

We list down all the modules that are used in multiple architectures throughout this study. The list consists of -

1. Encoders: We use two 1D-CNN encoders with slightly different structures; namely Encoder-1(for 34s segments) and Encoder-2(for 10s segments).
2. An FC Classifier to get final softmax scores.
3. An *Attention* module for LSTM based models. We use attention to calculate contributions of outputs at all time-steps to the final output and then use these weights to get a fixed length output vector which is used for classification task.

The CNN encoders are used for feature extraction from raw signals. These feature extractors are used by all classifiers presented in this study.

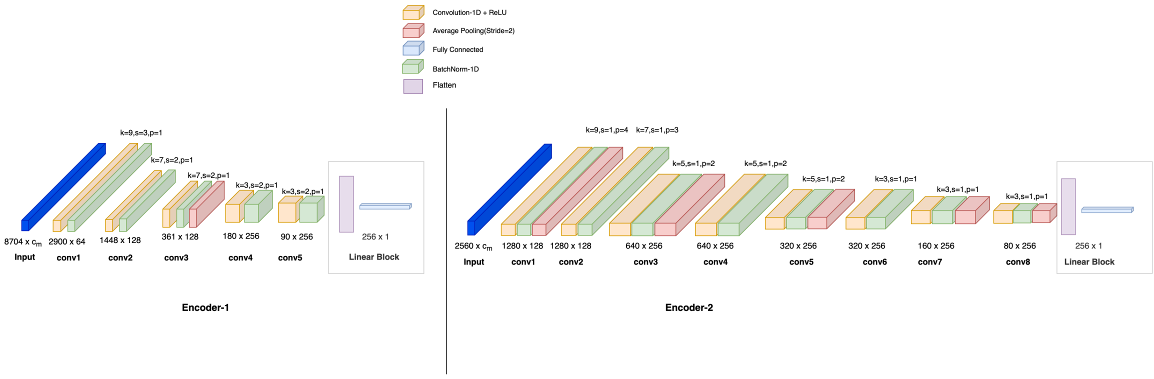


Figure 4.1: Encoder-1:CNN Encoder used for 34s and Encoder-2:CNN Encoder use for 10s segments. *Linear Block* is used to project the flattened output to a fixed size vector. The output shape of each layer is mentioned right below it.

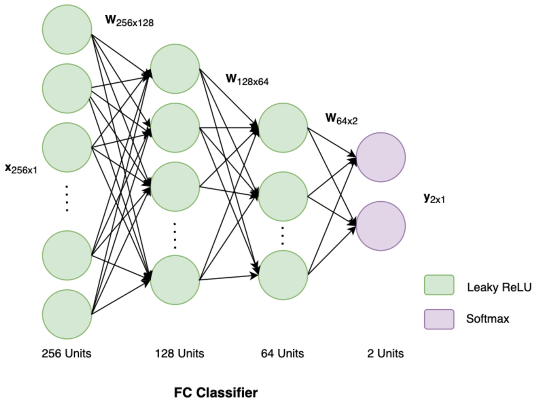


Figure 4.2: FC Classification head used for classification in all architectures.

Neural networks use attention to increase the focus on some parts of the input data while diminishing other parts. We briefly describe the general attention module below and then relate it to how we use this mechanism.

$$attention(\mathbf{K},\mathbf{q},\mathbf{V}) = similarity(\mathbf{K},\mathbf{q}).\mathbf{V} \tag{4.1}$$

$$similarity(\mathbf{K},\mathbf{q}) := weights = \frac{\mathbf{K}.\mathbf{q}}{\sqrt{d_{hidden}}} \tag{4.2}$$

$$attention(\mathbf{K},\mathbf{q},\mathbf{V}) := out = \mathbf{V}^T.weights \tag{4.3}$$

where:

Key Vectors := $\mathbf{K} \in \mathbb{R}^{T \times d_{hidden}}$
Value Vectors := $\mathbf{V} \in \mathbb{R}^{T \times d_{hidden}}$
Query Vector := $\mathbf{q} \in \mathbb{R}^{d_{hidden} \times 1}$
$weights \in \mathbb{R}^{T \times 1}$
$out \in \mathbb{R}^{d_{hidden} \times 1}$
T := Sequence Length
d_{hidden} := Hidden Representation Size

The Attention module is mainly used to get *fixed-length* representations for LSTM based models in this study. In our implementations the *Query Vector* is the *last hidden-state*(last time-step) of LSTM and the *Key* and *Value*Vectors are the sequence of hidden-states from all time-steps.

4.2 Experimental Settings and Evaluation Metrics

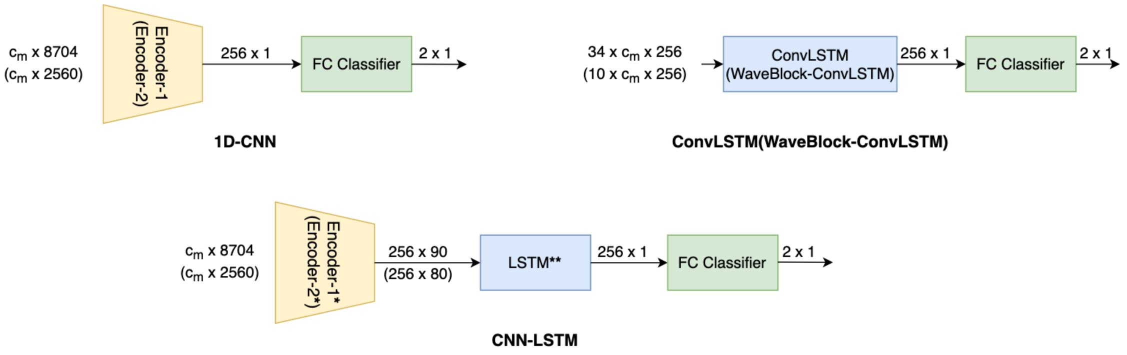
We train all the models for 15 epochs with starting learning rate of 0.0001. We use *Adam* optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$.
We use Cross-Entropy loss(see section-1.7.2) with a batch size of 64 for all the models.
We evaluate the models using a LOSO(see section-1.6) data split strategy, which yields 28 validation splits for MAHNOB-HCI dataset. We report mean and standard deviation of Macro Classification Accuracy, Macro F1-Score and ROC-AUC(see sections-[1.8.1, 1.8.2,1.8.3]).
We present the mean of all metrics along with their standard deviations calculated over all the 28 validation splits.

4.3 Unimodal Classification

For Unimodal classification(3.2.1) we use only one modality at a time. Starting with data preparation(Section-3.1), we use two different values of l_i(see see fig-3.1); 8704 and 2560(for 34s and 10s segments respectively) experiments.
The main motivation behind unimodal classification is to analyze how accurately and with how much certainty individual bio-signals predict emotional states.
In this section we describe the architectures we used for these experiments; the reasoning behind the choice of said architectures and corresponding results and observations.

4.3.1 Architectures

We list down all the architectures evaluated for unimodal classification in this study.



1D-CNN

ConvLSTM(WaveBlock-ConvLSTM)

CNN-LSTM

Figure 4.3: Input/Output sizes for various architectures. For ConvLSTM(WaveBlock-ConvLSTM) input is time sequence of length 34 and 10. *Encoder-1(Encoder-2) is without *Linear Block*. **Transpose before LSTM.

FC-Classifier

We start with a simple model. FC-Classifier is just the CNN encoders along with a 3-layer FC classification head. The classification metrics for valence and arousal classification are reported in tables-[4.1, 4.2].
The results show better performance for ECG and Resp signals than GSR and Temp. The classification performance is close to the results reported by [38].

Modality	F1-Score		Accuracy		ROC-AUC	
	10s	34s	10s	34s	10s	34s
ECG	61.54 ±8.62	60.49 ±9.96	62.47 ±7.83	61.99 ±8.32	59.93 ±8.27	59.66 ±11.08
GSR	51.88 ±9.81	58.13 ±10.20	56.58 ±7.54	60.53 ±8.02	55.51 ±11.97	61.20 ±10.67
Resp	55.92 ±7.82	60.43 ±10.49	57.87 ±6.76	62.86 ±8.14	57.86 ±9.42	59.67 ±12.58
Temp	55.11 ±8.63	59.09 ±6.59	57.37 ±7.39	60.57 ±6.77	53.56 ±10.38	58.72 ±11.51

Table 4.1: Valence Classification Results for FC-Classifier for MAHNOB-Data-V1 and MAHNOB-Data-V2

Modality	F1-Score		Accuracy		ROC-AUC	
	10s	34s	10s	34s	10s	34s
ECG	54.65 ±10.27	55.91 ±9.66	58.78 ±9.78	57.99 ±9.31	51.89 ±16.53	56.94 ±13.44
GSR	55.61 ±8.66	58.78 ±7.24	58.42 ±8.76	60.46 ±7.60	55.03 ±12.57	58.46 ±12.29
Resp	55.35 ±8.12	57.34 ±9.38	58.43 ±6.52	60.52 ±10.27	58.25 ±8.99	56.91 ±13.51
Temp	57.17 ±8.07	57.19 ±8.63	60.07 ±9.01	60.90 ±8.51	56.74 ±12.32	55.37 ±12.00

Table 4.2: Arousal Classification Results for FC-Classifier for MAHNOB-Data-V1 and MAHNOB-Data-V2

CNN-LSTM

The modalities considered in this study can be considered as scalar time series(for Resp, GSR and Temp) and multi-variate time-series in case of ECG. In order to incorporate this temporal property into the classifiers we use an LSTM model. The *Linear Block* from encoders is removed and replaced with an LSTM network. LSTM output is passed through an attention block to get a fixed length feature vector which is used as input for the classification head. The classification metrics for valence and arousal classification are reported in tables-[4.3, 4.4]. Once again the best results are achieved by ECG and Resp. Results for ECG and Resp are comparable to the corresponding results from FC-Classifier. However the F1-Scores for GSR and Temp drop-down significantly from the values reported for FC-Classifier. This effect on performances of GSR and Temp might be attributed to lack of any *frequency* information in these signals.

Modality	F1-Score		Accuracy		ROC-AUC	
	10s	34s	10s	34s	10s	34s
ECG	61.20 ±9.20	59.31 ±11.56	63.1 ±7.90	62.36 ±8.54	61.6 ±10.0	59.83 ±12.79
GSR	49.60 ±11.60	40.33 ±8.55	55.50 ±6.10	51.54 ±3.79	53.6 ±10.10	52.79 ±11.89
Resp	56.90 ±7.80	61.69 ±10.71	59.10 ±5.80	64.26 ±9.33	60.90 ±8.30	61.48 ±13.23
Temp	47.30 ±7.9	48.03 ±10.86	53.40 ±5.03	54.11 ±7.70	52.10 ±11.90	50.60 ±14.49

Table 4.3: Valence Classification Results for CNN-LSTM for MAHNOB-Data-V1 and MAHNOB-Data-V2

Modality	F1-Score		Accuracy		ROC-AUC	
	10s	34s	10s	34s	10s	34s
ECG	51.02 ±11.74	54.42 ±11.23	54.55 ±11.58	57.65 ±11.87	50.41 ±14.74	54.00 ±15.34
GSR	54.69 ±10.21	54.19 ±12.53	56.33 ±10.35	58.43 ±8.94	54.37 ±13.14	56.00 ±13.51
Resp	51.88 ±10.75	53.77 ±12.43	57.81 ±7.29	59.80 ±8.88	57.10 ±9.84	56.75 ±12.88
Temp	55.82 ±8.01	58.58 ±11.68	59.92 ±7.87	62.85 ±9.84	57.65 ±11.46	58.78 ±14.61

Table 4.4: Arousal Classification Results for CNN-LSTM for MAHNOB-Data-V1 and MAHNOB-Data-V2

ConvLSTM and WaveBlock-ConvLSTM

After recording the results for FC-Classifier and CNN-LSTM, we noticed that introducing LSTM deteriorates the performance which we did not expect.One of the reasons could be the lesser parameters in CNN-LSTM, another reason could be inefficient feature representations from CNN.

However we could not exactly determine any problem with the network itself. We decided to move ahead at this point and leave further analyses for the Analysis chapter.

We simply tried to extend the idea of CNN-LSTM to structures like ConvLSTM where we did not need a CNN encoder. In ConvLSTM the convolution happens at each time-step. We reshape the data bit as shown in Figure-4.3 and create a sequence of 1s segments and feed them through ConvLSTM.

For MAHBOB-Data-V1 the original shape of segments is $c_m \times 8704$ which is changed to $34 \times c_m \times 256$ and for MAHBOB-Data-V2 the original signal with shape $c_m \times 2560$ is changed to $10 \times c_m \times 256$

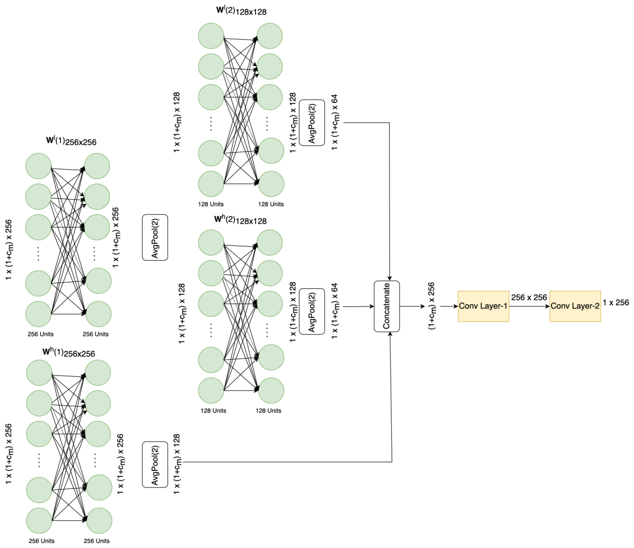


Figure 4.4: Illustration of structure and input/output-sizes of each gate of WaveBlock-ConvLSTM Cell for one time-step of the sequence. The number of channels in input at each timestep are $1 + c_m$ due to concatenation from previous hidden state which has 1 channel.

These models also employ the attention block to output a fixed-length feature vector which is used for final classification.

For WaveBlock-ConvLSTM we use the idea from WDNs(2.1.4) to extend the convolutions at each time step in ConvLSTM. In Figure-4.4 we show the WaveBlock-Conv structure of one gate which takes 1 time-step of sequence with shape $34 \times c_m \times 256$.

We report results on less number of experiments. We restricted the number of experiments to base minimum in order to first evaluate the improvement if any over previously reported results.

We suspect the poor results from ConvLSTM are mainly due to the data rather than the model itself. We try to describe these problems in detail in the *Analysis* chapter. WaveBlock-ConvLSTM suffers from the problem of parameters outburst, the number of parameters increases by a great margin even for 3 levels of wavelet decomposition. We report only limited results for these two models because of the limitations mentioned above.The results are reported in tables-[4.5, 4.6].

Modality	F1-Score	Accuracy	ROC-AUC
ECG	34.85 ±4.16	49.81 ±0.91	50.85 ±13.12
GSR	36.43 ±6.54	50.23 ±1.74	54.75 ±15.20
Resp	45.24 ±13.99	54.53 ±8.14	57.30 ±12.64

Table 4.5: Valence Classification Results for ConvLSTM only for MAHNOB-Data-V1

Modality	F1-Score	Accuracy	ROC-AUC
ECG	35.49 ±3.84	50.00 ±0.00	51.01 ±13.77
Resp	38.74 ±8.84	51.41 ±3.63	50.20 ±13.57

Table 4.6: Valence Classification Results for WaveBlock-ConvLSTM only for MAHNOB-Data-V1

4.4 Decision Fusion

Algorithm 1 Voting based Decision Fusion

```
Require: Set of unimodal classifiers $\mathcal{F}$
Require: $|\mathcal{F}| \geq 2$
  for $i < |\mathcal{D}_{test}|$ do
      $decision \leftarrow none$
      $high \leftarrow \{\}$                    ▷ Outputs of classifiers from $\mathcal{F}$ that predicted HA/HV
      $low \leftarrow \{\}$                     ▷ Outputs of classifiers from $\mathcal{F}$ that predicted LA/LV
      for $\mathcal{F}_m \in \mathcal{F}$ do
          $\hat{y}_i \leftarrow \mathcal{F}_m(m_i)$          ▷ $\hat{y}_i \in [0,1] \times [0,1] \subset \mathbb{R}^2$
          if $\text{argmax}\, \hat{y}_i == 1$ then
              $high \leftarrow high + \hat{y}_i$
          else
              $low \leftarrow low + \hat{y}_i$
          end if
      end for
      if $|high| > |low|$ then                  ▷ Majority vote to HA/HV
          $decision \leftarrow 1$
      else if $|high| < |low|$ then             ▷ Majority vote to LA/LV
          $decision \leftarrow 0$
      else                                      ▷ Prediction Confidence based Tie breaker
          $score\_high \leftarrow 0$             ▷ Total score achieved by HA/HV class
          $score\_low \leftarrow 0$              ▷ Total score achieved by LA/LV class
          $index \leftarrow 0$
          while $index < |high|$ do
              $h \leftarrow |high[index][1] - high[index][0]|$    ▷ Score for HA/HV class
              $score\_high \leftarrow score\_high + h$
              $l \leftarrow |low[index][1] - low[index][0]|$      ▷ Score for LA/LV class
              $score\_low \leftarrow score\_low + l$
              $index \leftarrow index + 1$
          end while
          if $score\_high > score\_low$ then
              $decision \leftarrow 1$
          else if $score\_high < score\_low$ then
              $decision \leftarrow 0$
          else
              $decision \leftarrow 1$
          end if
      end if
  end for
```

After recording results for various architectures for unimodal classification we use a simple voting based Decision Fusion(Section-1.10.1) to establish a baseline on all fusion methods evaluated in this study.

The algorithm uses the outputs $\mathbf{\hat{y}}_{im}$ from unimodal classifiers for modalities to be used for fusion. These outputs are then used to arrive at a consensus for the final result as seen in Equation-3.4.

The results for decision fusion for all modalities along with the models for which decision fusion was employed are listed in tables-[4.7, 4.8].

We can see improvements in FC-Classifier's valence classification accuracy and F1-score(see table-4.1) for MAHNOB-Data-V1(34s segments) while a drop in same metrics for MAHNOB-Data-V2(10s segments). However with increase in mean accuracy and F1-score value we also see an increase in standard deviation. Similar observation can be made for FC-Classifier's arousal classification accuracy and F1-score(see table-4.1) where mean values increase for 34s segments along with increase in standard deviation.

The decision fusion for CNN-LSTM model drops in case of both valence and arousal classification for MAHNOB-Data-V1 and MAHNOB-Data-v2.

Modality	F1-Score		Accuracy	
	10s	34s	10s	34s
FC-Classifier	59.26 ±8.33	61.85 ±12.28	60.88 ±7.09	63.54 ±9.81
CNN-LSTM	54.28 ±12.63	53.57 ±15.70	59.36 ±8.07	59.82 ±11.34

Table 4.7: Valence Classification Results from decision fusion of all 4 modalities for FC-Classifier and CNN-LSTM for MAHNOB-Data-V1 and MAHNOB-Data-V2

Modality	F1-Score		Accuracy	
	10s	34s	10s	34s
FC-Classifier	55.94 ±9.96	60.33 ±10.50	58.32 ±8.80	62.40 ±9.47
CNN-LSTM	52.11 ±12.12	54.08 ±12.36	57.37 ±10.17	60.33 ±11.56

Table 4.8: Arousal Classification Results from decision fusion of all the 4 modalities for FC-Classifier and CNN-LSTM for MAHNOB-Data-V1 and MAHNOB-Data-V2

4.5 Intermediate Fusion Pipeline

Intermediate Fusion is the main focus of this study. The motivation behind the idea is simple; with intermediate fusion we can really maximize the information we use during the *learning* process. To elaborate the point; for decision fusion we use the compressed information from multiple classifiers' outputs but we don't use the complete information while training the said classifiers; this idea gives intermediate fusion an edge over decision fusion.

In addition to above mentioned motivation, the voting based Decision Fusion does not perform much better than unimodal classifiers. The problem with decision fusion in this scenario is first and foremost the number of classifiers being used for fusion, since there are only 4 classifiers(ECG, GSR, Resp and Temp) combining or averaging probabilities(either with bagging or boosting) is not very effective.

There is one more problem with combining the results of these unimodal classifiers trained on MAHNOB-HCI; their outputs are very uncertain(Section-6.1) which leads to performance deterioration if results are averaged(combined).

The architectures evaluated for intermediate fusion are developed so that they extend unimodal classification architectures. The *Concatenation* and *Bilinear* fusion architectures are extensions

of *FC-Classifier* and similarly the *Cross-Modal Attention* architecture is basically a combination of two *CNN-LSTM* models with cross-attention.
The simplest all the methods we evaluate is the *Concatenation of modality representations* which as the same suggests is simple concatenation. We use this method along with decision fusion as baseline for other fusion methods.

4.5.1 Architectures

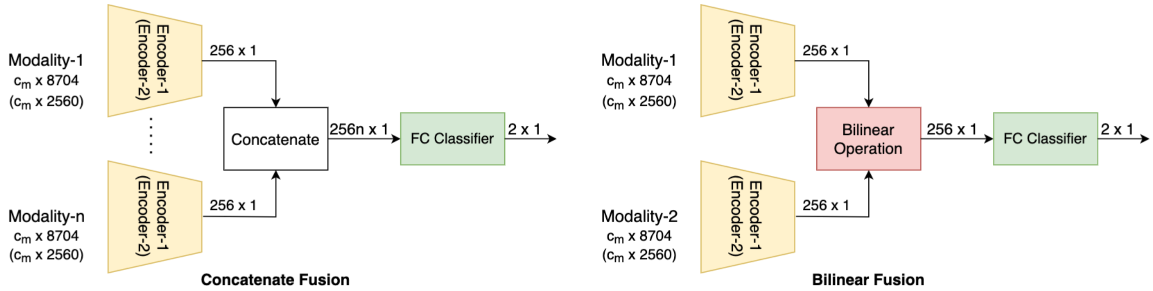


Figure 4.5: Concatenate Fusion and Bilinear Fusion Classifiers along with input/output shapes and sizes at various stages

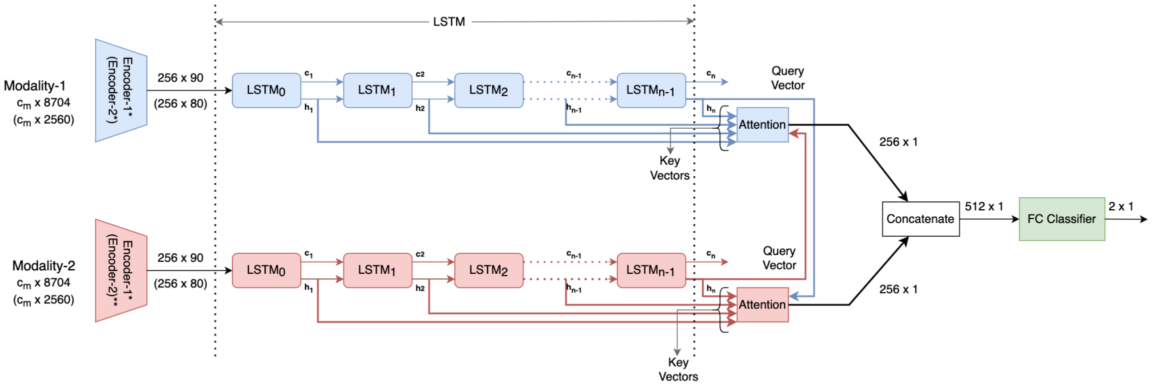


Figure 4.6: Cross-Modal Attention Classifier along with input/output shapes and sizes at various stages. Encoder output is transposed before it is used by LSTM. *Key* and *Value Vectors* from one modality along with *Query Vector* from other modality are used to calculate attention.

Concatenation of modality representations

Simply concatenate feature representations from encoders of multiple modalities for final classification as shown in Figure-4.5:Concatenate Fusion.
The results for concatenate fusion of ECG GSR and all 4 modalities are reported in tables-[4.9, 4.10]. We compare these results with those from FC-Classifier.
We see very slight improvement in the results for valence classification when all 4 modalities are concatenated with some exceptions in case of ROC-AUC example - in case of valence classification, ROC-AUC of concatenate fusion of all 4 modalities in lesser than ROC-AUC of all the modalities from FC-Classifier(see table-4.1).

For arousal classification the result is exactly the opposite, the performance for concatenate fusion drops as compared to individual modalities.
Comparing concatenate fusion results for ECG-GSR and all 4 modalities we can see that for valence classification all 4 modalities perform better; while for arousal classification ECG-GSR show better results.

Modality	F1-Score		Accuracy		ROC-AUC	
	10s	34s	10s	34s	10s	34s
ECG-GSR	60.36 ±7.99	60.66 ±8.58	61.25 ±7.16	62.32 ±7.34	60.04 ±9.47	58.77 ±11.39
All Signals	62.17 ±6.68	61.18 ±6.08	62.68 ±8.61	63.06 ±4.74	62.68 ±8.61	57.88 ±10.42

Table 4.9: Valence Classification Results for Concatenate Fusion for MAHNOB-Data-V1 and MAHNOB-Data-V2. Concatenation of all 4 modalities and ECG & GSR.

Modality	F1-Score		Accuracy		ROC-AUC	
	10s	34s	10s	34s	10s	34s
ECG-GSR	56.92 ±7.56	57.94 ±9.65	58.88 ±6.81	61.20 ±8.99	55.58 ±11.50	58.07 ±14.16
All Signals	54.20 ±10.04	56.92 ±9.14	58.93 ±8.14	60.54 ±9.22	56.52 ±13.62	58.65 ±14.77

Table 4.10: Arousal Classification Results for Concatenate Fusion for MAHNOB-Data-V1 and MAHNOB-Data-V2. Concatenation of all 4 modalities and ECG & GSR.

Bilinear Combination of modality representations

We explore multiple forms of interactions between modality feature-representation and *bilinear operation* as show in figure:4.5:Bilinear Fusion. The operation is defined as -

$$f(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T . \mathbf{W} \mathbf{y} \tag{4.4}$$

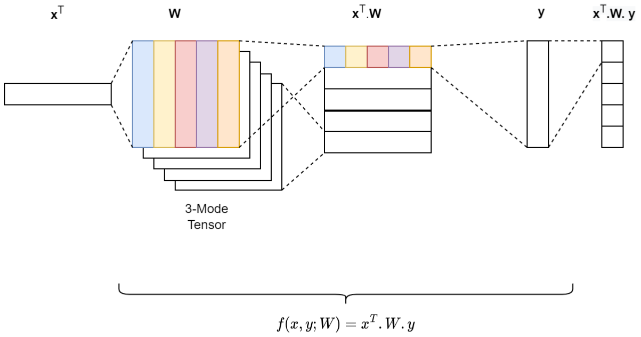


Figure 4.7: Illustration of Bilinear operation in network layer.

This method gets us the best results in terms of accuracy and also reduces uncertainty from unimodal classifiers(see section-6.1.2) This method has a limitation that only two modalities can be used at a time. The results for valence and arousal classification for bilinear fusion are reported in tables-[4.11, 4.12].
Comparing the results for bilinear fusion with individual modality results from FC-Classifier we see improvements in all metrics for both valence and arousal classification across both dataset versions.

Modality	F1-Score		Accuracy		ROC-AUC	
	10s	34s	10s	34s	10s	34s
ECG-GSR	62.02 ±7.72	64.48 ±7.97	63.49 ±6.61	65.44 ±7.18	63.05 ±10.83	62.54 ±10.25
ECG-Resp	62.31 ±6.15	62.90 ±7.55	63.17 ±6.04	63.91 ±6.82	64.44 ±9.92	62.20 ±11.11
Resp-Temp	59.83 ±6.67	63.67 ±9.75	61.34 ±6.16	65.13 ±9.46	63.04 ±9.06	61.84 ±12.23

Table 4.11: Valence Classification Results from Bilinear Fusion for both MAHNOB-Data-V1 and MAHNOB-Data-V2

Modality	F1-Score		Accuracy		ROC-AUC	
	10s	34s	10s	34s	10s	34s
ECG-GSR	58.96 ±7.86	67.38 ±9.91	61.80 ±8.11	68.70 ±10.03	62.01 ±10.12	65.95 ±14.88
ECG-Resp	61.68 ±5.99	63.01 ±8.16	63.87 ±6.48	66.02 ±8.51	63.48 ±10.83	63.89 ±14.13
Resp-Temp	60.82 ±6.74	63.71 ±9.06	63.09 ±6.07	66.23 ±8.16	65.91 ±7.52	64.86 ±10.55

Table 4.12: Arousal Classification Results from Bilinear Fusion for MAHNOB-Data-V1 and MAHNOB-Data-V2

Cross-Modal Attention

Cross-Modal attention is a very common method for aligning sequences of video and audio to localize events also the decoder attention the famous *Transformer* architecture where it uses this for *encoder-decoder attention*. We use the cross-attention method to fuse multiple modalities; we evaluate this method for two modalities at a time.The attention is calculated in the same way as mentioned in the *attention* description in Section 4.1 where we use sequence of hidden-states($\mathbf{h}_t$) from all time-steps as *Key Vectors* and *Value Vectors* and the last hidden-state($\mathbf{h}_n$) as the *Query Vector*. The only addition is that the query vector used now comes from the *second* modality as can be seen in Figure-4.6. The results are reported in tables-[4.13, 4.14]. When comparing these results with the results from CNN-LSTM model we see no performance improvements(except very few cases) instead the performance deteriorates. The reason for this may be bad alignment between segments from modality pairs.

Modality	F1-Score		Accuracy		ROC-AUC	
	10s	34s	10s	34s	10s	34s
ECG-GSR	59.42 ±10.46	55.87 ±12.68	61.67 ±8.89	59.46 ±9.04	61.57 ±11.30	55.43 ±11.12
ECG-Resp	59.84 ±9.07	58.29 ±10.38	62.38 ±6.93	61.18 ±8.09	62.15 ±9.13	56.64 ±9.23

Table 4.13: Valence Classification Results from Cross-Modal Attention Fusion for MAHNOB-Data-V1 and MAHNOB-Data-V2

Modality	F1-Score		Accuracy		ROC-AUC	
	10s	34s	10s	34s	10s	34s
ECG-GSR	53.63 ±12.04	53.71 ±10.22	57.49 ±11.12	58.32 ±10.42	54.87 ±15.43	52.62 ±17.05
ECG-Resp	54.22 ±10.13	52.31 ±12.93	58.74 ±8.03	58.52 ±9.45	56.88 ±13.54	54.29 ±14.03

Table 4.14: Arousal Classification Results from Cross-Modal Attention Fusion for MAHNOB-Data-V1 and MAHNOB-Data-V2

4.6 Results

Model	ECG (Valence)	ECG (Arousal)
FC-Classifier (Standalone)	62.00 ±8.3	57.9 ±9.3
CNN-LSTM (Standalone)	61.99 ±8.32	57.65 ±11.87
Concatenate (with all)	63.06 ±4.74	60.54 ±9.22
Concatenate (with GSR)	62.32 ±7.34	61.20 ±8.99
Bilinear (with GSR)	65.44 ±7.18	68.70 ±10.03
Bilinear (with Resp)	63.91 ±6.82	66.02 ±8.51
Cross-Modal (with GSR)	59.46 ±9.04	58.32 ±10.42
Cross-Modal (with Resp)	61.18 ±8.09	58.52 ±9.45

Model	GSR (Valence)	GSR (Arousal)
FC-Classifier (Standalone)	60.5 ±8.0	60.5 ±7.6
CNN-LSTM (Standalone)	60.53 ±8.03	58.43 ±8.94
Concatenate (All)	63.06 ±4.74	60.54 ±9.22
Concatenate (with ECG)	62.32 ±7.34	61.20 ±8.99
Bilinear (with ECG)	65.44 ±7.18	68.70 ±10.03
Cross-Modal (with ECG)	59.46 ±9.04	58.32 ±10.42

Table 4.15: Final results for ECG and GSR

Model	Resp (Valence)	Resp (Arousal)
FC-Classifier (Standalone)	62.9 ±8.1	60.5 ±10.3
CNN-LSTM (Standalone)	62.86 ±8.14	59.80 ±8.88
Concatenate (All)	63.06 ±4.74	60.54 ±9.22
Bilinear (with ECG)	63.91 ±6.82	66.02 ±8.51
Bilinear (with Temp)	65.13 ±9.46	66.23 ±8.16
Cross-Modal (with ECG)	61.18 ±8.09	58.52 ±9.45

Model	Temp (Valence)	Temp (Arousal)
FC-Classifier (Standalone)	60.6 ±6.8	60.9 ±8.5
CNN-LSTM (Standalone)	60.58 ±6.78	62.85 ±9.84
Concatenate (All)	63.06 ±4.74	60.54 ±9.22
Bilinear (with Resp)	65.13 ±9.46	66.23 ±8.16

Table 4.16: Final results for Resp and Temp

5 Pretraining of the Bio-Signal Feature Extractor

After collecting results from models trained end-to-end as classifiers; we note that there is still a lot of room for improvement. We hypothesize that small number of samples in the dataset along with the presence of inter-subject variations, combined with LOSO testing strategy makes it difficult to get good results.

We attempt to solve the two problems mentioned above i.e. the inter-subject variations and smaller dataset size by pretraining the feature extractor for bio-signals in unsupervised fashion.

Unsupervised training enables us to skip LOSO; since we only model the data without any labels we can use data of all the subjects; or even the data coming from new subjects over time. This helps solve the problem of inter-subject variations when learning feature representations.

Now to address the problem of less data for training the feature extractor and start with our solution we present a more detailed breakdown of MAHNOB-HCI dataset. MAHNOB-HCI dataset in total has 2 type of experiments *Emotion Elicitation* and *Tagging*. The self-rating of valence and aroual is available only for trials coming from the Emotion Elicitation; so in total we have 547 trials with valence-arousal labels. However the other trials from tagging experiment don't have such labels but come from same subjects and are conducted in the same conditions.

Experiment	Stimuli	Stimuli Durations	# of Trials
Emotion Elicitation	Videos	34s - 120s	547
Tagging	Videos and Images	10s - 14s	2392

Table 5.1: Breakdown of MAHNOB-HCI dataset.

We hypothesize that if we use signals from all the trials combined to pretrain our feature extractor before the final classifier training we might alleviate the problem of small dataset, as far as feature representations are concerned.

There is one downside in using all the trials. The more trials we use the more risk we run of running into outliers. This becomes a major problem when we normalize the data using min-max normalization.

5.1 Data

We use all the trials(Emotion Elicitation + Tagging) in MAHNOB-HCI dataset with preprocessing settings similar to MAHNOB-Data-V2(Table-3.3) for training the autoencoders.

5.2 Problem Formulation

Give a set of segments for each modality -

$$\mathcal{S}_{im} \in \mathcal{D}_m \quad \forall \quad m \in \mathcal{M} \tag{5.1}$$

We want to learn a parametrized function $f_{\theta m}$ such that

$$f(\mathcal{S}_{im};\theta_m) = \hat{\mathcal{S}}_{im} \quad \ni \quad \hat{\mathcal{S}}_{im} \approx \mathcal{S}_{im} \tag{5.2}$$

5.3 Model

The autoencoder model consists of *Encoder-2* module and a corresponding decoder.
The decoder consists of *transposed convolutions* to upsample the downsampled signal through pooling layers in the encoder. Each transposed convolution layer is also followed by *ReLU* activation.

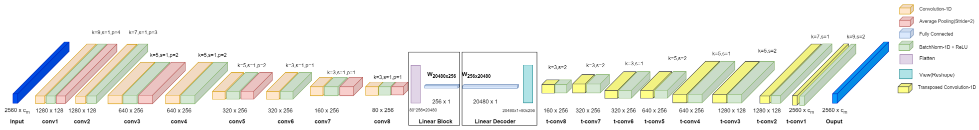


Figure 5.1: Autoencoder model with all the layers in order. The output shape of each stage is mentioned below it along with the name of the stage. The input and output have the same shape.

5.4 Training and Evaluation

The autoencoder is trained using NVIDIA GPU for 25 epochs. We use *Adam* Optimizer with a learning rate of 0.001 and step-wise learning rate scheduler.
We use MSE as the loss metric for autoencoder training. We keep 2 trials from each subject as test data and rest is used for training the autoencoder.

After finishing the autoencoder training, we use the encoder(everything to the left of including the Linear Block from Figure-5.1) to repeat unimodal classification experiments(sec-4.3) with *FC-Classifier*(4.3.1) to compare with the original results obtained.
When training the classifiers we set a learning rate of 0.0001 for the encoder and 0.001 for the rest of the classifier. The reasoning was to not update the weights of encoder completely which may cause catastrophic forgetting. We also experimented with finetuning only the last layers of encoder while classifier training but got better results with the lower learning rate setup. We repeat the classification experiment using LOSO testing strategy with 5 randomly selected subjects and report the results.

5.5 Results

We start with ECG signal and report reconstruction loss over the course of training and then move on to compare the classification results with the encoder taken from this trained autoencoder.

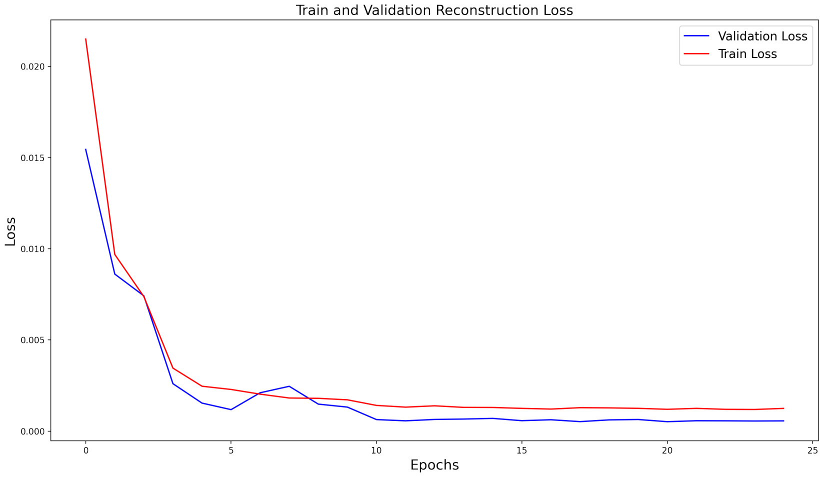


Figure 5.2: Trends of reconstruction loss over the course of training for 25 epochs for autoencoder with ECG data.

Subject-Id	Accuracy (Encoder from Scratch)	Accuracy (Pretrained Encoder)
8	45.83	51.78
21	65.73	63.13
17	63.78	58.59
14	69.94	64.58
19	51.88	57.89

Table 5.2: Comparison of FC-Classifier with encoder trained from scratch and the pretrained encoder. The ECG data used for this comparison comes from MAHNOB-Data-V2 i.e. the 10s segments.

As we can see from Table-5.2 the accuracy with the pre-trained encoder is lower is most cases however there are few cases for which this accuracy increases for example subject-8 and subject-19.

In conclusion pretraining the encoder with this method does not bring us the desired results i.e. the overall classification accuracy does not increase and the standard deviations of accuracy over the subjects also does not decrease. One of the possible reasons for this behaviour might be the presence of outliers which disturb the normalization process, the other reason could be the underlying data and the inter-subject deviations that we mention in the beginning of this chapter and in section-6.3.1, where we mention the extent of such deviations briefly.
Since we did not see any major improvements we would not focus on this method further in this study.

6 Analysis

In the previous chapters we conclude the classification experiments with multiple models. We also considered pretraining for the feature extractor in order to improve the performance. Despite observing improvements using bilinear fusion, the results are not promising. Hence in this chapter we analyse the possible reasons behind the less promising results.

We see significant changes in outputs for a given sample depending on the initialization of the model. To further study this observation is more detail we perform uncertainty analysis for the simplest models we have i.e. the *FC-Classifier*.

We also visualize raw signals from trials based on what rating they have received and which stimulus was used for the said trials along with handcrafted feature analysis and latent feature analysis from multiple perspectives.

6.1 Uncertainty Analysis

In the classification results we see similar performance across models and also dataset versions along with varying results for multiple random initialization. We hypothesize that this uncertainty mainly comes from how our dataset is structured. We set out to analyse segments from raw signals and how our model reacts to them, in order to narrow down our search space we go through the following steps -

1. Initialize the same model with multiple random seeds and train these different versions.
2. Run inference on all samples to get softmax outputs; accumulate these outputs across model versions from Step-1.
3. Measure *predictive entropy* for each sample.
4. Sort the samples in increasing order of uncertainty and pick n samples from bottom and from the top.
5. Next we analyse these two groups of samples to check for properties.

Predictive Entropy[[39], [16]] has its root from Information Theory and can be used measure uncertainty in model predictions(modelling both epistemic and aleatory uncertainty). This method mainly uses *Deep Ensembles*, set of same deep-models with differing initialization. Uncertainty is usually measured for test data, however we measure it for both test and train data. Since with different initialization the fit on train data is also very different and brings out different results. Given our main goal of narrowing down the number of samples to study in raw form, train data is a better choice.

$$H[y|x, \mathcal{D}_{train}] = -\Sigma_c p(y|x). \log p(y|x, D_{train}) \tag{6.1}$$

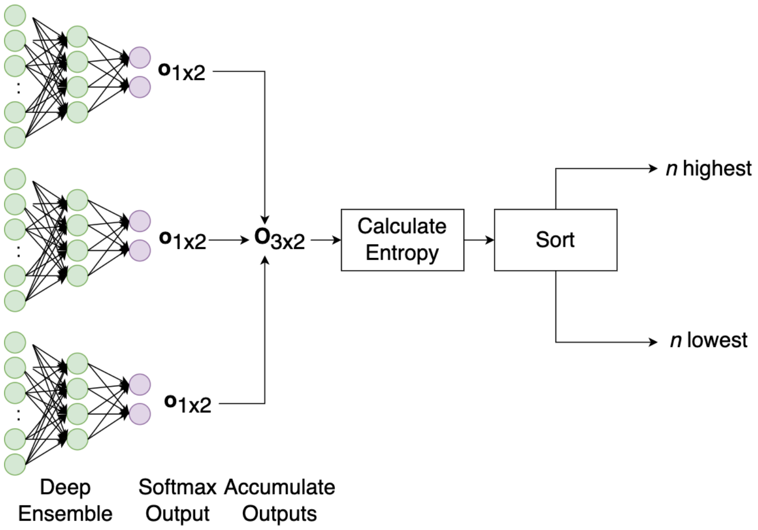


Figure 6.1: Pipeline for narrowing down on samples to analyze models' reaction on through Deep Ensemble and Predictive Entropy

We use the predictive entropy measurements on test data as an additional metric, along with accuracy and ROC-AUC scores to measure performance of fusion methods.

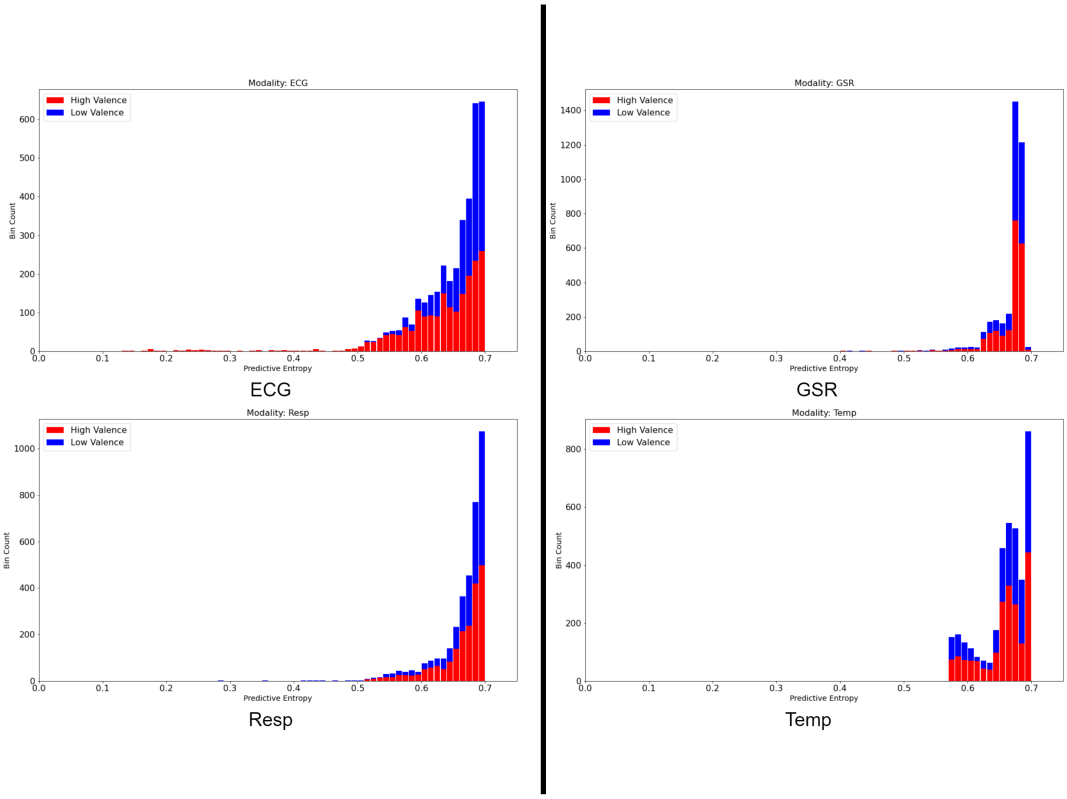
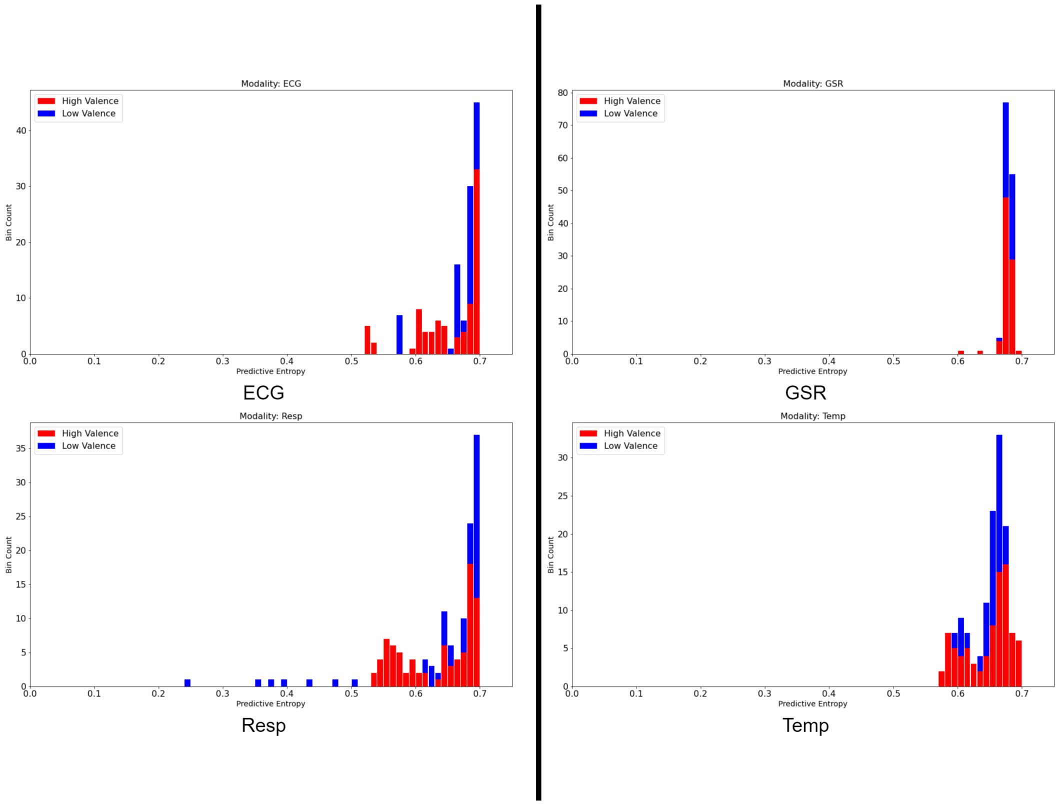


Figure 6.2: Histograms for predictive entropy for ECG, GSR, Resp and Temp for training data with Subject-8 left out as test subject from MAHNOB-Data-V2(3.3)



ECG

GSR

Resp

Temp

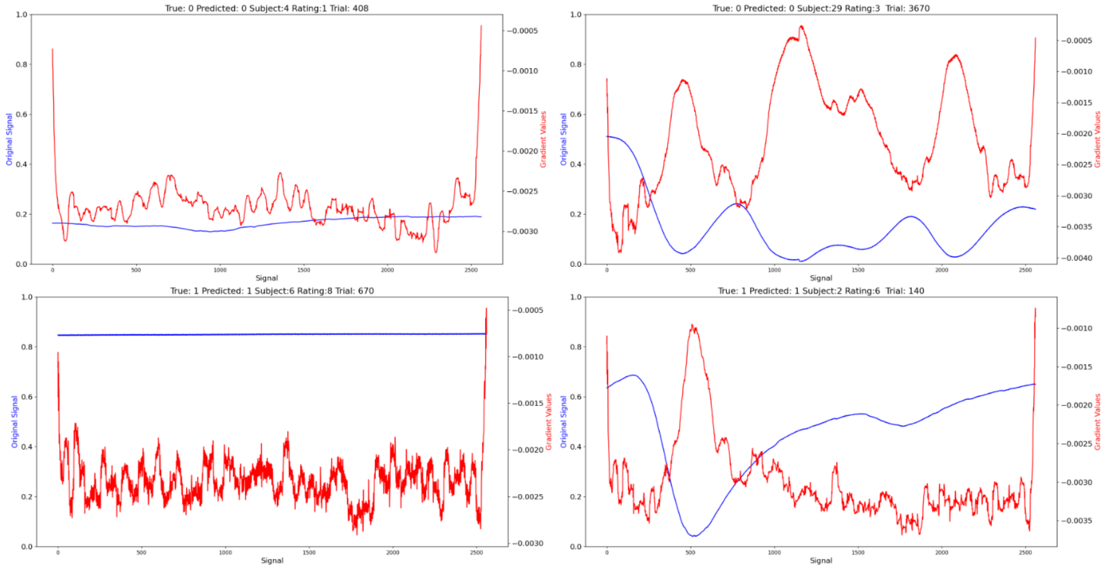
Figure 6.3: Histograms for predictive entropy for ECG, GSR, Resp and Temp data for test data with Subject-8 as test subject from MAHNOB-Data-V2(3.3)

The histograms from figures-[6.2, 6.3] are for Valence Classification for subject-8. We see *bins* of predictive entropy on x-axis and bin count on y-axis. We can see how train and test data both exhibit high entropy for most of the samples. This is the main reason behind the poor results from the classification experiments. *Covariate Shift* in training data could be one of the possible reasons of these high values of predictive entropy for samples in the training data. We wish to analyse this shift in a little detail in later sections in this chapter.
These plots also show that ECG and Resp have more samples in low entropy regions when compared to GSR and Temp. This information combined with accuracy and f-score measures confirms that ECG and Resp are better predictors of emotion intensity; also confirmed by [38].

6.1.1 Samples from extreme ends

Out main goal of measuring predictive entropy on train data was to narrow down our analyses space for raw signals. As depicted in Figure-6.1 we take n samples with highest and n with lowest values of entropy. These samples come from training data(not test data) since we want to study raw segments.
We then measure gradient attributions for these extreme groups of samples through method of *Integrated Gradients*; this was as a side experiment to visualize how model reacts to samples from the extreme ends of predictive entropy distribution. We present some the samples for each modality along with the gradient attributions.

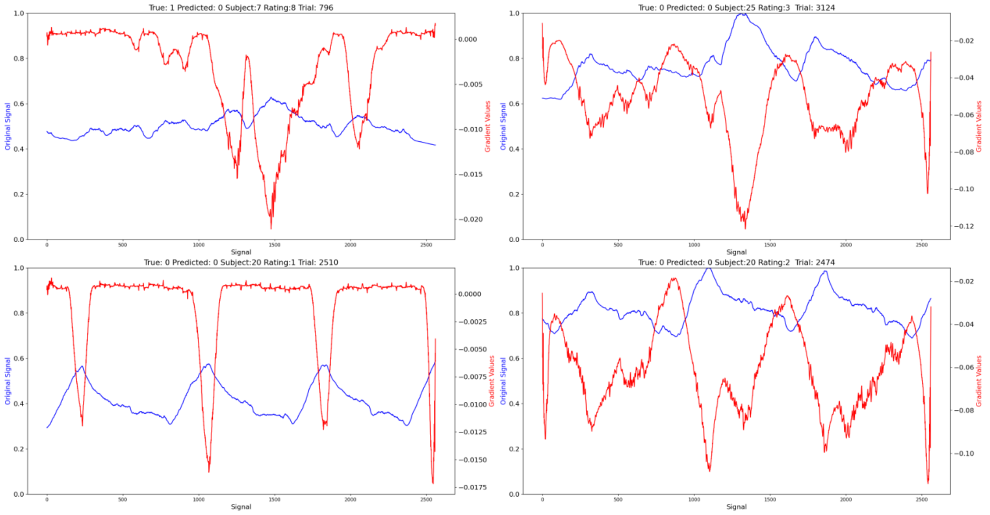
This analysis is done using MAHNOB-Data-V2 where the segments are 10s long. The GSR and Resp signals are in blue along with their respective gradient attributions in red.



High Entropy Samples Low Entropy Samples

Figure 6.4: Extreme cases - Highest and lowest entropy samples for GSR(Blue) along with gradient attributions(Red).

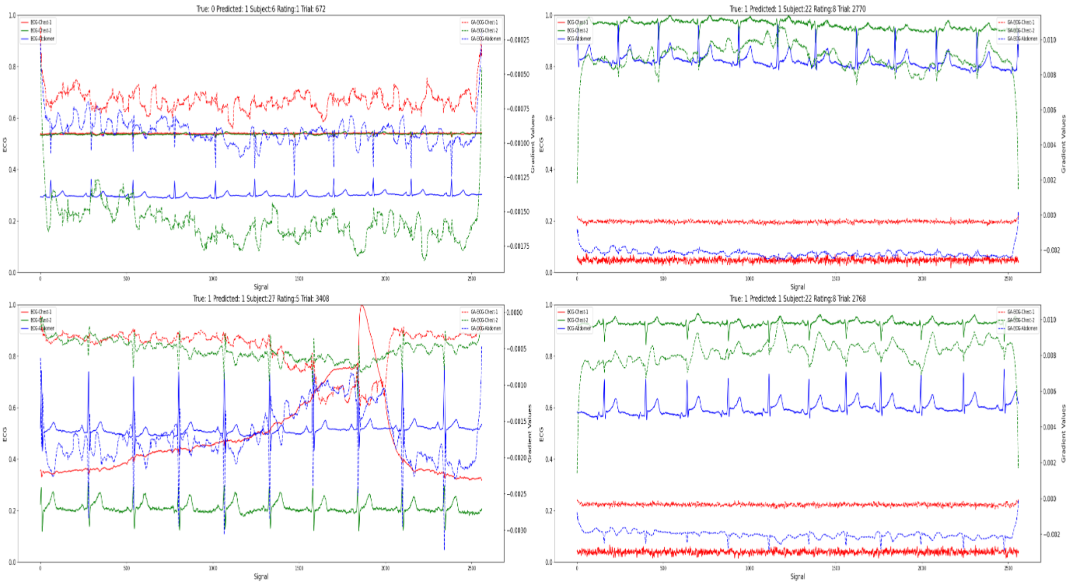
In Figure-6.4 we can see extreme cases for GSR. Samples on the left seem to have these characteristic non-fluctuating signals which essentially represent absence of any information. This is the reason we use longer segments from each trial as well i.e. *MAHNOB-Data-V1* where we see slight improvements in the results.



High Entropy Samples Low Entropy Samples

Figure 6.5: Extreme cases - Highest and lowest entropy samples for Resp(Blue) along with gradient attributions(Red).

The situation is somewhat different for Resp where we see lower amplitudes in samples with high predictive-entropy compared to samples that exhibit lower predictive-entropy. This again is a point of concern since the model should not focus on amplitude for Resp rather frequency of peaks and valleys.



High Entropy Samples

Low Entropy Samples

Figure 6.6: Extreme cases - Highest and lowest entropy samples for ECG Signals(solid lines) and corresponding gradient attributions(dashed lines)

Similar plots for ECG have a slightly structure, owing to the multiple channels. We have the signals from three channels in solid lines and their corresponding gradient attributions as dashed lines of same color.

Sample analysis of ECG shows results more difficult to understand with the interesting property that all low entropy samples come from subject-22 but different trials. Where the solid-red line(ECG Chest sensor) seems to be always malformed. These samples very likely are malformed or are outliers distorted by the normalization process, here we see only a part of the complete trial.

We do not do similar analysis for Temp for primarily two reasons -

1. The temperature in MAHNOB-HCI dataset is taken from the tip of the small finger which is not very representative of body core temperature. For example for subject-11 the range of temperature 23-25 Degree Celsius.

2. As can be seen in Section-6.2 and Figures 6.9 and 6.10 that there is significant bias in the results when we ignore the baseline.

6.1.2 Uncertainty after fusion

We can treat uncertainty as measured by predictive entropy as an additional metric to compare model fits . We calculate these uncertainty histograms after fusion and compare

them with individual modalities. We see that fusion helps in reducing predictive entropy and is robust towards network initialization than individual modalities.
In the figures the Fusion data comes from Bilinear fusion classifier and individual modalities' data from *FC-Classifier*
We see in Figures-[6.7, 6.8] that the more samples lie in the low entropy region for fusion method as compared to individual modalities. This observation is true for train and test data both.

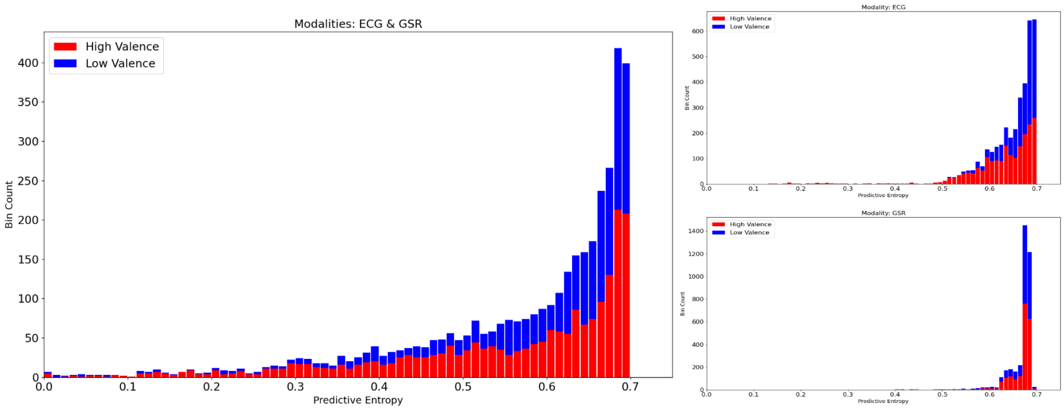


Figure 6.7: Histograms for predictive entropy for Bilinear fusion of ECG GSR along with individual signals(smaller images on the right) for training data with Subject-8 left out as test subject from MAHNOB-Data-V2(3.3)

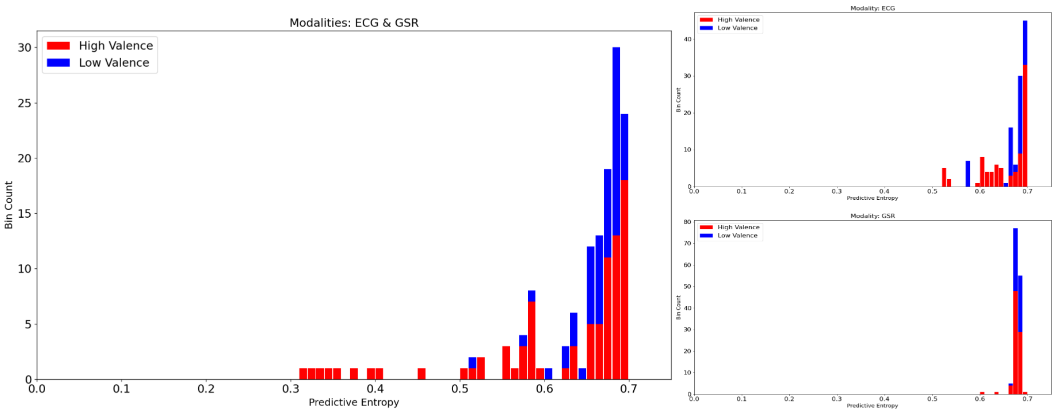


Figure 6.8: Histograms for predictive entropy for Bilinear fusion of ECG GSR along with individual signals(smaller images on the right) for test data with Subject-8 as test subject from MAHNOB-Data-V2(3.3)

6.2 Baseline vs Stimulus Periods

In this section we redirect our focus of analysis of signals from the perspective of stimuli. We do this exercise with the goal to find out if there is any correlation between observed signals

to the stimuli that caused them across subjects.
To achieve this goal we do multiple types of comparisons for example visualizing/analysing the raw signals ans their frequency responses. We also use hand-crafted features and so t-SNE visualization to study the effects of such handcrafted features in stimulus period vs baseline period.

6.2.1 Raw Signals

In this section we study raw signals. Since Temp is the easiest signal to study owing to very low frequency and monotonous nature of temperature changes in human skin, we start with this signal.

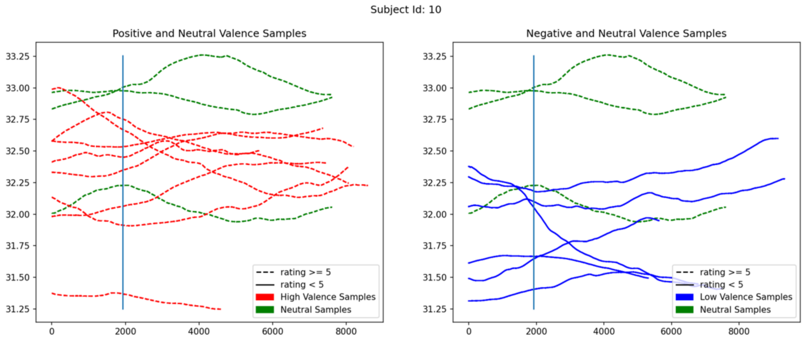


Figure 6.9: Skin temperature plots of subject-10 when subjected to high valence stimuli in red(Joy and Amusement), low valence stimuli in blue(Sadness and Disgust) and neutral stimuli in green along with the self-rating, shown by dashed lines if rating >= 5 and solid lines for rating < 5. Vertical line represents start of stimulus.

In the above figure raw temperature values from the dataset are plotted for trials coming from reactions to high and low valence samples; the idea behind these plots is to visibly see the effect of such stimulus on the subject and how their skin temperature changes.
For subject-10 we make an interesting observation that baseline temperatures for high valence stimuli are higher than those in low valence stimuli; now since we ignore the baseline in our experiments we make the classifier vulnerable to overfit to this decision boundary of difference in amplitudes which is not a desired effect.

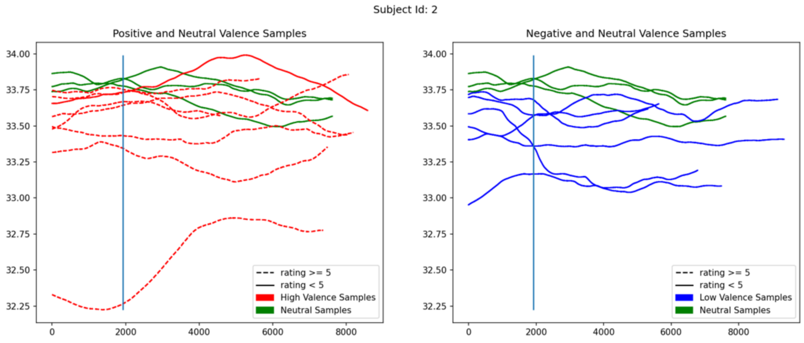


Figure 6.10: Skin temperature plots of subject-2

We see the effect mentioned above and ROC of *FC-Classifier* for subject-10 and modality Temp is quite high compared to other subjects with a value of 67.40 while for subject-2 where these amplitudes don't differ a lot the ROC of Temp modality falls drastically to a value of 52.33.

Overall we could not see any common patterns across the two subjects that relate the skin-temperatures recorded with the nature of stimulus or their self-rating to the trial.
The plots a little different for Resp and GSR since they are not as smooth as Temp plotting them the same way as Temp in Fig-6.9 makes the plot very chaotic.

Since Respiration Amplitude is also a frequency based signal we plot the frequency spectrum along with original and de-trended signals. The frequency spectrum of Resp are computed after removing trend and the filtering using Butterworth LPF with a cutoff frequency of 10Hz. The the figures below for subjects-10 and subject-16 show that there is not enough frequency information to distinguish between the ratings with naked eye. Also we don't see much correlation between signals recorded for different ratings, preliminary emotion and respiration amplitudes.

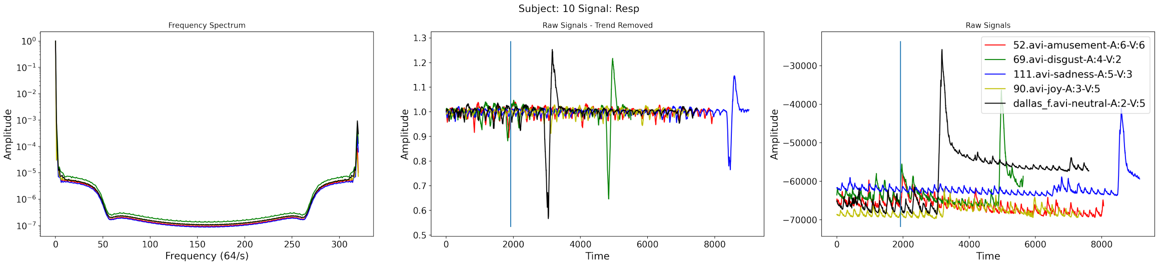


Figure 6.11: Respiration signals for multiple stimulus from different baseline emotions for subject-10. The frequency spectrum is on the left, raw signals on right with trend removed signals in the middle. The vertical line marks end of baseline

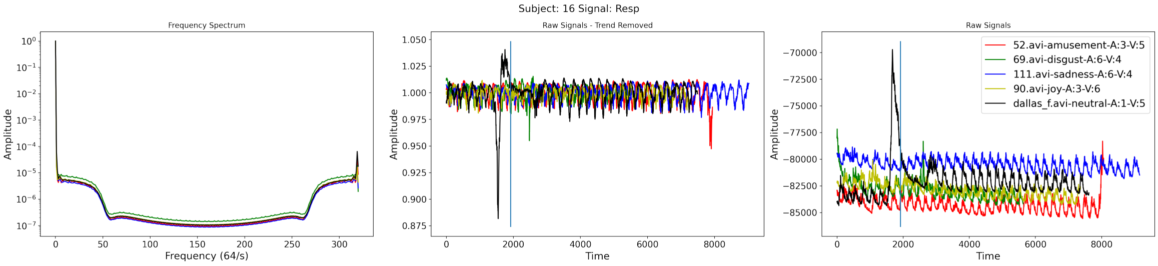


Figure 6.12: Respiration signals for multiple stimulus from different baseline emotions for subject-16. The frequency spectrum is on the left, raw signals on right with trend removed signals in the middle. The vertical line marks end of baseline

The difference in mean amplitude of various trials may be due to the movement of the respiration belt along the abdomen.

For GSR we do not present any frequency domain analysis. In the following two figures we point out that in many cases the variation in GSR signal does not show very strong correlation the rating given by participant. For example in case of subject-10 both high and low arousal ratings have this unusual characteristic of almost no variations. While for subject-16 most of the trials irrespective of the stimulus or rating have this trend of one big trough and subsequent upward trend in the signal.

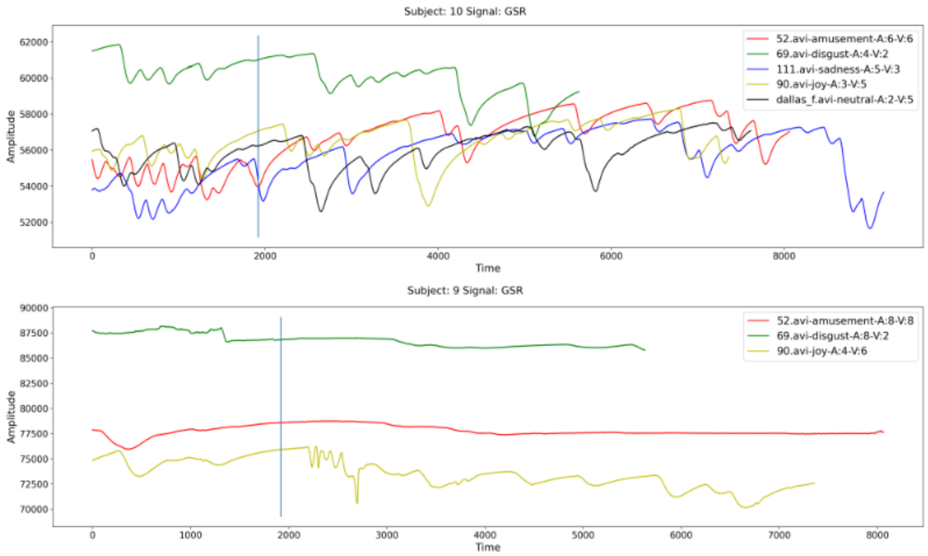


Figure 6.13: GSR signals for multiple stimulus from different baseline emotions for subjects 9 and 10. The vertical line marks end of baseline

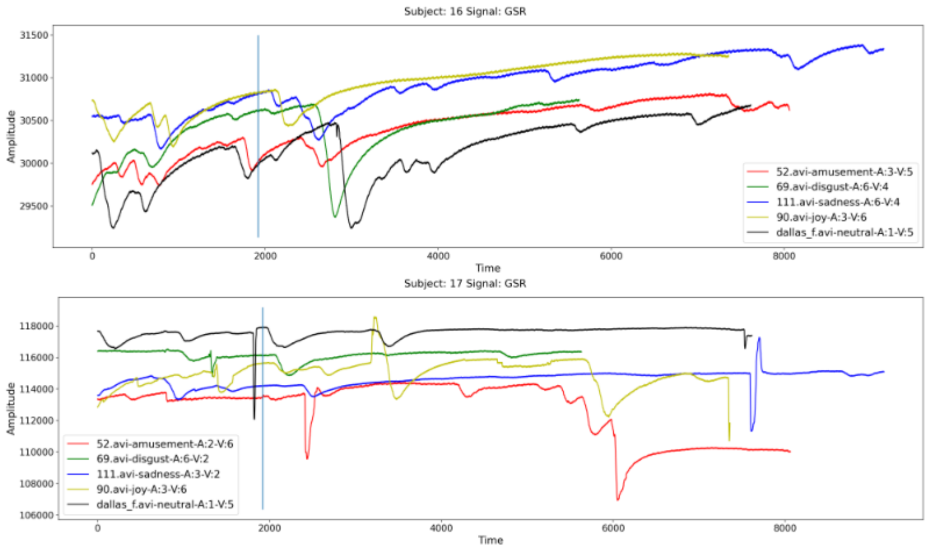


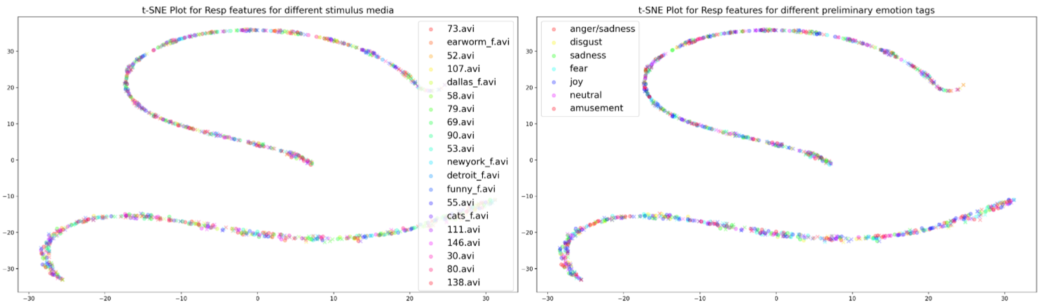
Figure 6.14: GSR signals for multiple stimulus from different baseline emotions for subjects 16 and 17. The vertical line marks end of baseline

6.2.2 Hand-Crafted Features

We also analyze hand-crafted features used by [40] from the perspective of stimuli and also with the goal to see how different the baseline and stimulus periods are, as we did with the raw signals. First we separate each trial into baseline and stimulus periods; then we use handcrafted features for both periods to create a feature vector for each. Once we have all the feature vectors we do a t-SNE visualization of the data.

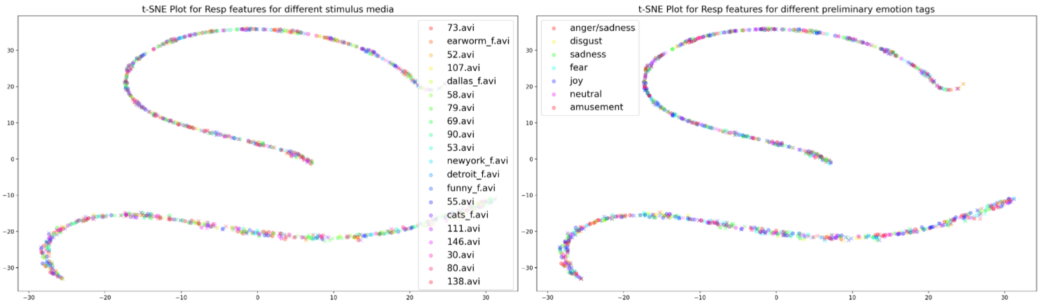
For Resp we take the statistical features i.e. minimum, maximum, mean, median, variance and maximum-minimum difference for both time and frequency domain. Also we calculate the first and second difference of the Resp signal in time domain and take minimum, maximum, mean, median, variance and maximum-minimum difference. This gives us the feature vector for Resp signal for each trial of both baseline and stimulus periods.

We repeat the same procedure for GSR. The goal is to be able to see the see the difference between baseline and stimulus feature vectors. We do a 2-component t-SNE visualization of both of these feature vectors in the following figures.



(a) (b)

Figure 6.15: t-SNE visualization of hand-crafted features for Resp signal. The colors denote the preliminary emotion tag and media for the trial and ’x’ marks for baseline features and ’o’ marks for stimulus features



(a) (b)

Figure 6.16: t-SNE visualization of hand-crafted features for GSR signal. The colors denote the preliminary emotion tag and media for the trial and ’x’ marks for baseline features and ’o’ marks for stimulus features

We see no visible difference between feature vectors for stimulus or baseline periods. Also we cannot observe any correlation between feature vectors for stimulus periods and the associated media or emotion tags.

We also note another problem with using features like maximum, minimum, mean and median for both Resp and GSR, as can been from seen from Figures-[6.11, 6.12], these features are not correlated with the self-rating; the overall vertical shift that we see in the raw signals will affect these features a lot, which is not representative of any stimulus but rather can be associated to movement of the sensory belt along the abdomen. We highly doubt that such features, especially for MAHNOB-HCI dataset can predict emotional intensity with much accuracy.

6.3 Latent Features Analysis

As mentioned in the previous section, we find that the handcrafted features such as minimum, maximum, mean and median are not suitable for our analyses because they are much more affected by external factors rather the stimuli.
We wish to overcome this challenge by using latent feature representations instead of such hand-crafted features for further analyses. We could use the outputs from encoders used in classifiers but we did the want the bias in features coming from learning a decision boundary. We again turn to unsupervised learning and train a Variational Autoencoder introduced in section-1.5 to generate latent representations for segments from MAHNOB-Data-V2.
The representations generated from VAE gives two advantages -

1. A proper probabilistic structure of the latent space.
2. VAE models the data not the decision boundary.

The generated latent representations $\mathbf{z}$ from the encoder are then used. The latent space has a multivariate diagonal normal distribution as mentioned below.

$$\mathbf{z} \sim \mathcal{N}(\mu, \Sigma); \quad \mu \in \mathbb{R}^{256}, \; \Sigma \in \mathbb{R}^{256\times 256}$$

Since the distribution is diagonal; meaning no covariance across dimensions in latent space, each dimension can be treated independently.

6.3.1 Analyses of Latent Subgroups

After generating latent representations from VAE we primarily use them for analysing subgroups in the dataset from the multiple perspectives. We wish to confirm the presence of any such subgroups from the perspective of stimuli, subjects and the self-rating for valence and arousal.
Any dominant sub-groups(for example subject-level) which are conditionally independent of the class labels might explain co-variate shifts in training data leading to unstable training and high uncertainty in predictions
The presence of such groups is identified using a distance metric between distributions of these groups. We use *Wasserstein Metric*[[41], [42]] to measure this distance. We calculate this distance for each dimension in latent space(owing to diagonal distribution) to get a vector $\mathbf{d}_{1\times 256}$. The final distance d is calculated as -

$$d = \frac{||\mathbf{d}_{1\times 256}||_2}{256} \tag{6.2}$$

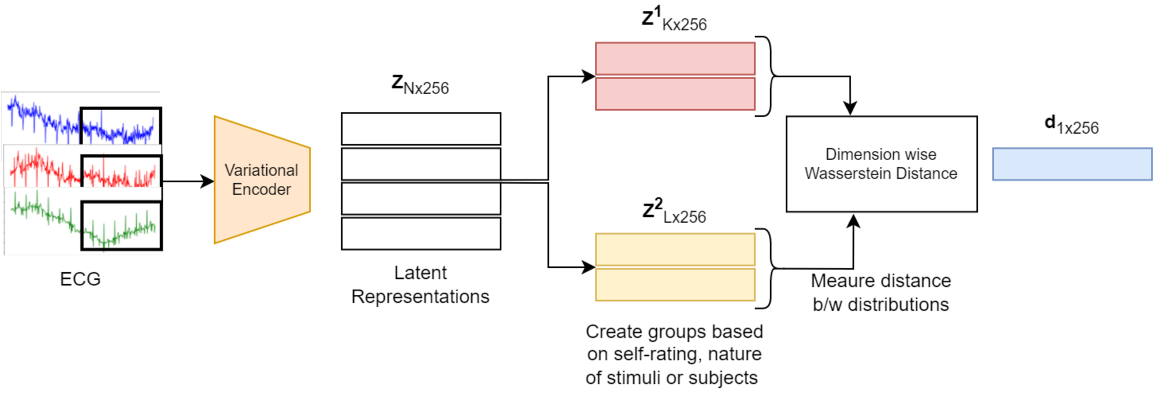
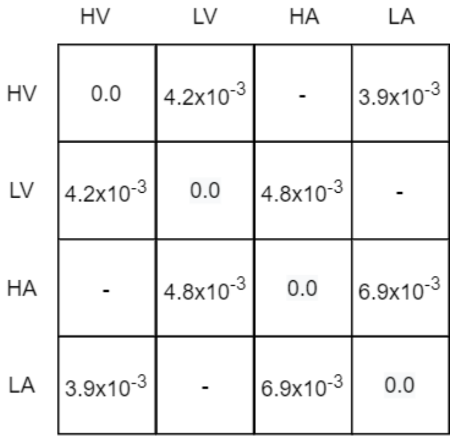

z^1_{Kx256}
Z_{Nx256}
Variational Encoder
ECG
Latent Representations
Dimension wise Wasserstein Distance
d_{1x256}
z^2_{Lx256}
Meaure distance b/w distributions
Create groups based on self-rating, nature of stimuli or subjects

Figure 6.17: Illustration of process used to calculate the distance between the groups as described above

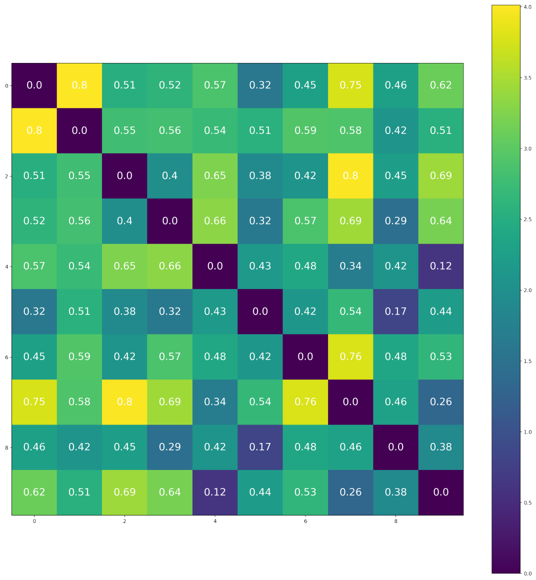
Starting with ECG latent features, the first parameter we select is *self-rating*. We divide the data into 4 groups i.e. HV, LV, HA and LA based on self rating and measure the described distance amongst these groups taken two a time.



	HV	LV	HA	LA
HV	0.0	4.2×10^{-3}	-	3.9×10^{-3}
LV	4.2×10^{-3}	0.0	4.8×10^{-3}	-
HA	-	4.8×10^{-3}	0.0	6.9×10^{-3}
LA	3.9×10^{-3}	-	6.9×10^{-3}	0.0

Figure 6.18: Wasserstein Distances calculated for HA, LA, HV, LV taken two at a time. We did not calculate the distances for pairs denoted by '-'

In order to check of how inter-class distances for standard datasets look like , we also report wassterstein distance metric for MNIST Dataset[43] where separate classes are treated as groups.

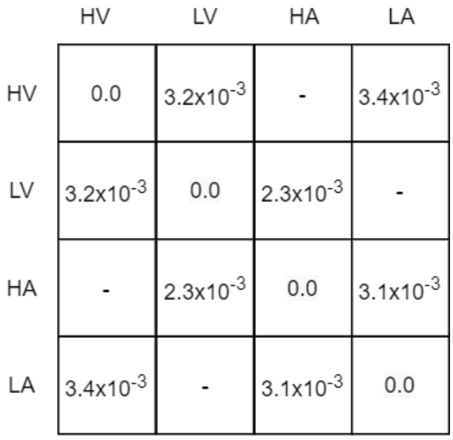


0.0	0.8	0.51	0.52	0.57	0.32	0.45	0.75	0.46	0.62
0.8	0.0	0.55	0.56	0.54	0.51	0.59	0.58	0.42	0.51
0.51	0.55	0.0	0.4	0.65	0.38	0.42	0.8	0.45	0.69
0.52	0.56	0.4	0.0	0.66	0.32	0.57	0.69	0.29	0.64
0.57	0.54	0.65	0.66	0.0	0.43	0.48	0.34	0.42	0.12
0.32	0.51	0.38	0.32	0.43	0.0	0.42	0.54	0.17	0.44
0.45	0.59	0.42	0.57	0.48	0.42	0.0	0.76	0.48	0.53
0.75	0.58	0.8	0.69	0.34	0.54	0.76	0.0	0.46	0.26
0.46	0.42	0.45	0.29	0.42	0.17	0.48	0.46	0.0	0.38
0.62	0.51	0.69	0.64	0.12	0.44	0.53	0.26	0.38	0.0

Figure 6.19: Wasserstein Distances calculated for distributions of 10 classes of MNIST Data with the same method

The minimum value of inter-class distance for MNIST is 0.12 much greater than those between HA-LA and HV-LV.

We also divide the data and create two groups for high/low valence stimulus - where amusement, joy video clips are treated as high valence stimuli and sadness, fear and disgust as low valence(Section-1.9.3). We repeat the same procedure for high/low arousal and measure the distance.



	HV	LV	HA	LA
HV	0.0	3.2×10^{-3}	-	3.4×10^{-3}
LV	3.2×10^{-3}	0.0	2.3×10^{-3}	-
HA	-	2.3×10^{-3}	0.0	3.1×10^{-3}
LA	3.4×10^{-3}	-	3.1×10^{-3}	0.0

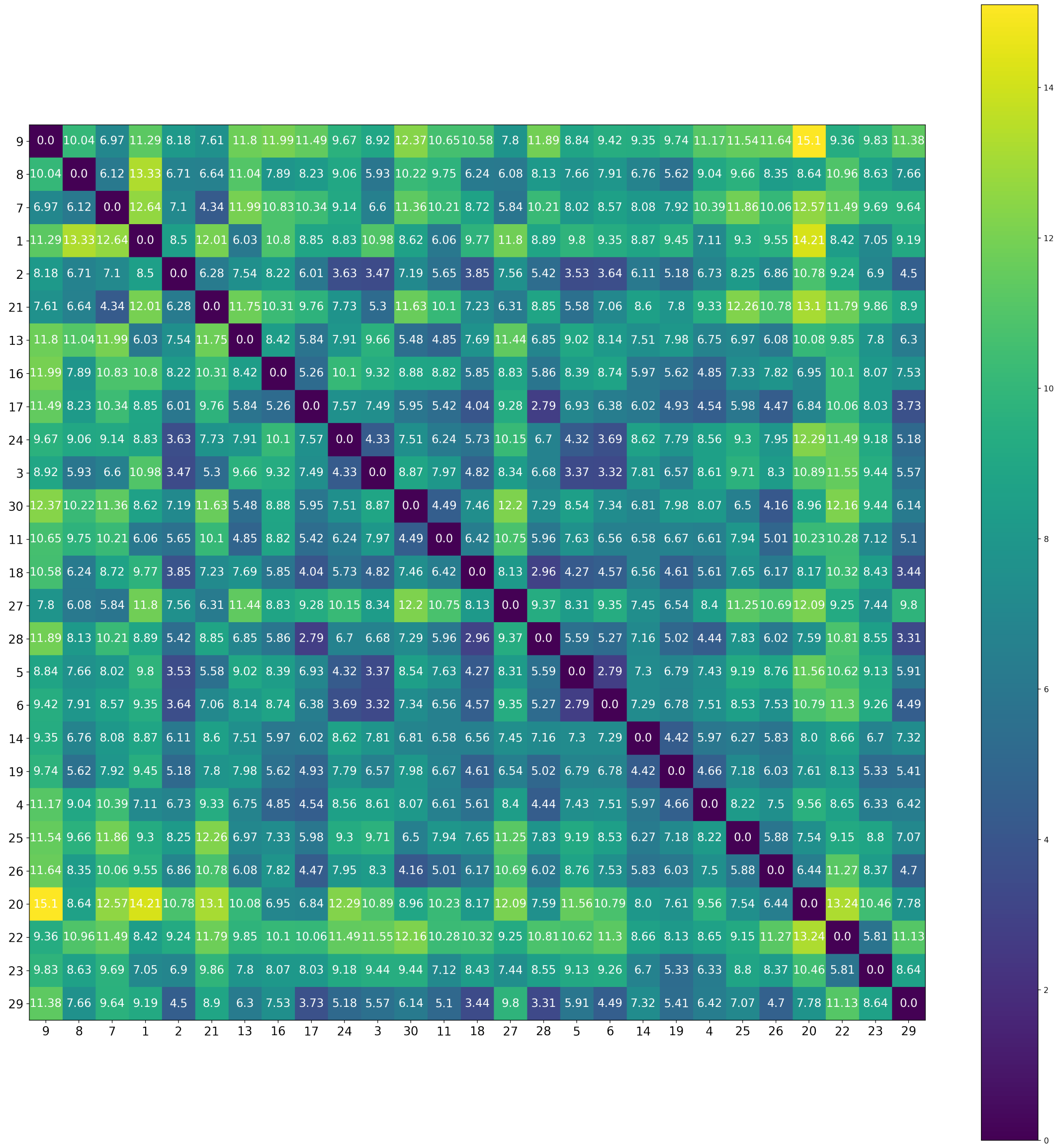
Figure 6.20: Wasserstein Distances calculated for HA, LA, HV, LV Stimuli taken two at a time. We did not calculate the distances for pairs denoted by '-'

The values are comparable to those for the rating groups.

Next, we treat data from each subject as one group. We measure the distances between all subject groups, taken 2 at a time.

In Figure-6.21 it is clear that the inter-subject differences($31.3\times10^{-3}\pm9.01\times10^{-3}$ - across all subject pairs) as shown by the latent representations are much greater than inter-class(HA-

LA or HV-LV) differences or High/Low Valence or Arousal stimulus differences.
With this analysis we get some level of confirmation about the claims made regarding high inter-subject variance and ineffectiveness of handcrafted features like minimum, maximum values etc.
We also make a very important observation here; if our analysis of these latent features is correct, we don't see much shifts between distributions for high and low self-ratings but large shifts between data from different subjects. This could explain the sub-par classification results. We might also consider these subject level groups as an indicator of co-variate shifts in the training data which to such high uncertainty observed in the classifiers(see section-6.1)



	9	8	7	1	2	21	13	16	17	24	3	30	11	18	27	28	5	6	14	19	4	25	26	20	22	23	29
9	0.0	10.04	6.97	11.29	8.18	7.61	11.8	11.99	11.49	9.67	8.92	12.37	10.65	10.58	7.8	11.89	8.84	9.42	9.35	9.74	11.17	11.54	11.64	15.1	9.36	9.83	11.38
8	10.04	0.0	6.12	13.33	6.71	6.64	11.04	7.89	8.23	9.06	5.93	10.22	9.75	6.24	6.08	8.13	7.66	7.91	6.76	5.62	9.04	9.66	8.35	8.64	10.96	8.63	7.66
7	6.97	6.12	0.0	12.64	7.1	4.34	11.99	10.83	10.34	9.14	6.6	11.36	10.21	8.72	5.84	10.21	8.02	8.57	8.08	7.92	10.39	11.86	10.06	12.57	11.49	9.69	9.64
1	11.29	13.33	12.64	0.0	8.5	12.01	6.03	10.8	8.85	8.83	10.98	8.62	6.06	9.77	11.8	8.89	9.8	9.35	8.87	9.45	7.11	9.3	9.55	14.21	8.42	7.05	9.19
2	8.18	6.71	7.1	8.5	0.0	6.28	7.54	8.22	6.01	3.63	3.47	7.19	5.65	3.85	7.56	5.42	3.53	3.64	6.11	5.18	6.73	8.25	6.86	10.78	9.24	6.9	4.5
21	7.61	6.64	4.34	12.01	6.28	0.0	11.75	10.31	9.76	7.73	5.3	11.63	10.1	7.23	6.31	8.85	5.58	7.06	8.6	7.8	9.33	12.26	10.78	13.1	11.79	9.86	8.9
13	11.8	11.04	11.99	6.03	7.54	11.75	0.0	8.42	5.84	7.91	9.66	5.48	4.85	7.69	11.44	6.85	9.02	8.14	7.51	7.98	6.75	6.97	6.08	10.08	9.85	7.8	6.3
16	11.99	7.89	10.83	10.8	8.22	10.31	8.42	0.0	5.26	10.1	9.32	8.88	8.82	5.85	8.83	5.86	8.39	8.74	5.97	5.62	4.85	7.33	7.82	6.95	10.1	8.07	7.53
17	11.49	8.23	10.34	8.85	6.01	9.76	5.84	5.26	0.0	7.57	7.49	5.95	5.42	4.04	9.28	2.79	6.93	6.38	6.02	4.93	4.54	5.98	4.47	6.84	10.06	8.03	3.73
24	9.67	9.06	9.14	8.83	3.63	7.73	7.91	10.1	7.57	0.0	4.33	7.51	6.24	5.73	10.15	6.7	4.32	3.69	8.62	7.79	8.56	9.3	7.95	12.29	11.49	9.18	5.18
3	8.92	5.93	6.6	10.98	3.47	5.3	9.66	9.32	7.49	4.33	0.0	8.87	7.97	4.82	8.34	6.68	3.37	3.32	7.81	6.57	8.61	9.71	8.3	10.89	11.55	9.44	5.57
30	12.37	10.22	11.36	8.62	7.19	11.63	5.48	8.88	5.95	7.51	8.87	0.0	4.49	7.46	12.2	7.29	8.54	7.34	6.81	7.98	8.07	6.5	4.16	8.96	12.16	9.44	6.14
11	10.65	9.75	10.21	6.06	5.65	10.1	4.85	8.82	5.42	6.24	7.97	4.49	0.0	6.42	10.75	5.96	7.63	6.56	6.58	6.67	6.61	7.94	5.01	10.23	10.28	7.12	5.1
18	10.58	6.24	8.72	9.77	3.85	7.23	7.69	5.85	4.04	5.73	4.82	7.46	6.42	0.0	8.13	2.96	4.27	4.57	6.56	4.61	5.61	7.65	6.17	8.17	10.32	8.43	3.44
27	7.8	6.08	5.84	11.8	7.56	6.31	11.44	8.83	9.28	10.15	8.34	12.2	10.75	8.13	0.0	9.37	8.31	9.35	7.45	6.54	8.4	11.25	10.69	12.09	9.25	7.44	9.8
28	11.89	8.13	10.21	8.89	5.42	8.85	6.85	5.86	2.79	6.7	6.68	7.29	5.96	2.96	9.37	0.0	5.59	5.27	7.16	5.02	4.44	7.83	6.02	7.59	10.81	8.55	3.31
5	8.84	7.66	8.02	9.8	3.53	5.58	9.02	8.39	6.93	4.32	3.37	8.54	7.63	4.27	8.31	5.59	0.0	2.79	7.3	6.79	7.43	9.19	8.76	11.56	10.62	9.13	5.91
6	9.42	7.91	8.57	9.35	3.64	7.06	8.14	8.74	6.38	3.69	3.32	7.34	6.56	4.57	9.35	5.27	2.79	0.0	7.29	6.78	7.51	8.53	7.53	10.79	11.3	9.26	4.49
14	9.35	6.76	8.08	8.87	6.11	8.6	7.51	5.97	6.02	8.62	7.81	6.81	6.58	6.56	7.45	7.16	7.3	7.29	0.0	4.42	5.97	6.27	5.83	8.0	8.66	6.7	7.32
19	9.74	5.62	7.92	9.45	5.18	7.8	7.98	5.62	4.93	7.79	6.57	7.98	6.67	4.61	6.54	5.02	6.79	6.78	4.42	0.0	4.66	7.18	6.03	7.61	8.13	5.33	5.41
4	11.17	9.04	10.39	7.11	6.73	9.33	6.75	4.85	4.54	8.56	8.61	8.07	6.61	5.61	8.4	4.44	7.43	7.51	5.97	4.66	0.0	8.22	7.5	9.56	8.65	6.33	6.42
25	11.54	9.66	11.86	9.3	8.25	12.26	6.97	7.33	5.98	9.3	9.71	6.5	7.94	7.65	11.25	7.83	9.19	8.53	6.27	7.18	8.22	0.0	5.88	7.54	9.15	8.8	7.07
26	11.64	8.35	10.06	9.55	6.86	10.78	6.08	7.82	4.47	7.95	8.3	4.16	5.01	6.17	10.69	6.02	8.76	7.53	5.83	6.03	7.5	5.88	0.0	6.44	11.27	8.37	4.7
20	15.1	8.64	12.57	14.21	10.78	13.1	10.08	6.95	6.84	12.29	10.89	8.96	10.23	8.17	12.09	7.59	11.56	10.79	8.0	7.61	9.56	7.54	6.44	0.0	13.24	10.46	7.78
22	9.36	10.96	11.49	8.42	9.24	11.79	9.85	10.1	10.06	11.49	11.55	12.16	10.28	10.32	9.25	10.81	10.62	11.3	8.66	8.13	8.65	9.15	11.27	13.24	0.0	5.81	11.13
23	9.83	8.63	9.69	7.05	6.9	9.86	7.8	8.07	8.03	9.18	9.44	9.44	7.12	8.43	7.44	8.55	9.13	9.26	6.7	5.33	6.33	8.8	8.37	10.46	5.81	0.0	8.64
29	11.38	7.66	9.64	9.19	4.5	8.9	6.3	7.53	3.73	5.18	5.57	6.14	5.1	3.44	9.8	3.31	5.91	4.49	7.32	5.41	6.42	7.07	4.7	7.78	11.13	8.64	0.0

Figure 6.21: Wasserstein Distances between distribution of latent vectors for subjects from MAHNOB-HCI Dataset

7 Discussion

In this study we evaluated multiple architectures for unimodal classification and then fusion methods. Although the fusion methods evaluated by us show an improvement over unimodal classification we wanted to dig a little deeper as we noticed a number of problems.
The first major problem we found was high uncertainty in results for same model due to network initialization(Section-6.1); for some cases we could easily identify the problems like for GSR the segmentation process leads to a lot of flat, no-information regions for 10s segments. We try to alleviate this problem with 34s segments at the cost of total number of samples for classification. This exercise also helps use identify some trials with unexpected ECG signals; this is a major problem since the data is already very small. We also try to reduce the uncertainty by using a pretrained encoder as a feature extractor for bio-signals but we could not achieve the desired accuracy.
The other major problem is the fact that different subjects have very high differences in results for any given modality. This can be seen in the high standard deviations in all the classification results. We hypothesize that this problem arises from some unexpected behavior of signals for example in case Temp Modality in Figures-[6.9, 6.10] we see major performance improvement for subject-10; however if we analyze the signals there is nothing explaining such performance improvement apart from the fact that after removing the baseline period it becomes very easy to separate high vs low rating samples for subject-10. If the baseline were considered and changes with respect to baseline were treated as an indicator for emotional intensity we would not see such performance improvement. The same might be the case for other subjects with other modalities.
We would also like to point out that the process of *Segmentation*(fig-3.1) is sort of a hack. The main reason of doing segmentation is to make more data out a single trial, which is required for effective training of deep learning models. However we have seen in this work that this sometimes lead to completely empty(of information) segments which lead to a bad training process. If we imagine a scenario that a lot of segments are not very representative of the self-rating and since these segments are part of single trial so we can assume large correlation among them; then if result for one of them goes wrong the result for rest goes wrong. We tried to correct this by using a single large segment of 34s from trial end and it does improve the results a little but its not full-proof, because this reduces the number of datapoints drastically and increases the dimensionality of the segment vector. This remains an open problem.
The next set of problems draw our attention when we try to analyse the results to figure out the reasons for above mentioned problems. In order to check if there is any actual difference between baseline and stimulus periods, we plot raw signals and hand-crafted features for multiple trials(Figures-[6.11, 6.13, 6.15, 6.16]) and we could not see any difference between the said period for any of raw signals or handcrafted features.

Using the handcrafted features we also try check for any correlation between stimulus and corresponding signals.This observation provides some explanation for the poor classification performance.

Although we use these handcrafted features to make the above observation; just like we point out at the end of Section-6.2.2 these features might be not very good representations of any actual effect of stimulus. Notice in Figures-[6.9, 6.11, 6.14] for raw signals the the signals show vertical shifts along the amplitude axis(baseline included); which is an effect of respiration belt movement along abdomen for Resp and similar external effects for other modalities. These vertical shifts will change the maximum-minimum values to a very large extent, in-turn changing the feature vectors without any relation to self-rating.

This argument against trustworthiness of the handcrafted features motivated us to try and analyse latent representations that might eliminate any direct impacts of these external effects by modelling the data distribution directly.
We analyse the distributions of these latent representations to see any patterns resulting from stimuli and self-rating. We did not see any major distribution changes for features from high self-rating to low self-rating. The biggest shifts we see is between subject based latent sub-groups in the dataset.
These presence of such sub-groups support our doubts of inter-subject variations as mentioned in Chapter-5. The presence of such groups in this low data classification task along with the fact that these groups are *conditionally independent* from the associated class labels also explains the uncertainty due to different network initialization. We hypothesize that these groups introduce the problem of co-variate shifts while the training process and there is not enough data to overcome this problem which makes the results very much dependent on the initialization.
We provide the details all of the trials that we found unusual in the Appendix(Sections-[10.4, 10.4]). We had a goal of making the problem of emotion detection very simple through an end-to-end system; but we acknowledge that such a task requires very powerful filtering for removal of any such wanted artifacts. We mention the next steps possible in the priority we would want to tackle them in next chapter.

8 Future Works

After the observing the results from classification experiments and the analyses done during the during the course of this study, we believe the first next step that should be prioritized over all else is a deeper dive into the MAHNOB-HCI dataset.

We base this prioritization on the fact that we found some unexpected data in trials for ECG without even looking for them explicitly in the Uncertainty Analysis section(6.1). MAHNOB-HCI is a decade old dataset and is quite small so these abnormalities were not expected by us.

Studying the dataset in detail and rooting out any such errors would be crucial for any type of future work with this dataset.

As a next step we should focus on learning and taking into account this inter-subject variability; almost none of the works including our own tries to take this variance into consideration while solving the main classification problem. We believe that unsupervised and semi-supervised learning might help with learning representations robust to this variance.

Also the way we and many other works[[13], [44]] deal with the problem of small size of MAHNOB data is via segmentation. This helps alleviate the problem of very few labelled samples; however since the segments do not contain the complete information and also could be completely information less as can be seen in *High Entropy Samples* for GSR in Figure-6.4, this method does not come without a cost.

This problem becomes very prominent when coupled with the idea that emotion is localized at certain points in the trial making all other segments redundant and they may cause performance degradation. This problems can lead to another very important direction for future work which can focus on proper data selection from the trials that does not produce the same problems as Segmentation does. The work done by [45] along the same lines is a good starting point.

9 Conclusion

In this study we tackle the problem of emotion classification by formulating it as a binary classification problem for emotion intensities.
We use end-to-end deep learning approaches for the classification task where we evaluate effectiveness of single modalities namely ECG, GSR, Resp and Temp and also fusion of these modalities.
Fusion methods outperform any single modality. However we face many other problems which we try to analyse starting from raw signals from data along with other ways to see the impact of stimuli on the peripheral signals for different participants.
We conclude by mentioning once more that although we got close to state-of-the methods following LOSO validation split; we wish to analyze and possibly fix the problems related to this emotion classification task in our future works.

10 Literature Research

10.1 Deep Learning based Fusion Methods

- „Utilizing Deep Learning Towards Multi-modal Bio-sensing and Vision-based“ [12]
    - Results for individual modalities along with fusion results.
    - Use of pretrained VGG-16 network to extract features from PSD images from EEG signals.
    - For ECG/PPG signals; the signals are converted in spectrograms and again a pretrained VGG-16 is used to extract features.
    - No to very little information on how the features were *combined* together.
    - Validation with 10-Fold Cross Validation with 80/20 split; LOSO not employed.
- „DeepVANet: A Deep End-to-End Network for Multi-modal Emotion Recognition“ [13]
    - Fuse video and bio-signals at feature level.
    - Bio-Signals fused in an early fusion style as channels of 1D-Spatial signals.
    - 1D-Convolutions and LSTM as modelling units.
    - Mention of per-subject evaluation - Similar to LOSO; Data leakage found on trial level.
- „Deep Learning Method for Selecting Effective Models and Feature Groups in Emotion Recognition Using an Asian Multimodal Database“ [45]
    - Uses Genetic Algorithm for selecting models given the dataset.
    - LSTM model used for each EEG channel and simple concatenation of final LSTM outputs are fed to FCN.
    - LOSO not employed.
    - The authors got continuous emotional state tagging results from MAHNOB-HCI authors. This could be a good idea and might solve the problem of localization of emotion that we face.
- „Multimodal Emotion Recognition Using a Hierarchical Fusion Convolutional Neural Network“ [46]
    - Handcrafted features extracted from EEG and PPS(PPG, GSR, Resp Temp) individually.
    - The preprocessed EEG and PPS signals formed in a unified vector and then used as input for CNN to extract features.
    - Handcrafted features and convolutional features fused together with weighted fusion.
    - Random Forest as final classifier. LOSO not employed.

- „Automatic Emotion Recognition Using Temporal Multimodal Deep Learning“ [47]
  - EEG and BVP signals used.
  - Signals have individual CNN encoders followed by concantenation of CNN output at each time-step t. Author’s call this their *Early Fusion*.
  - They also present an alternative where the modalities have individual CNN-LSTM encoders and the final output is concatenated for classification and this is represented as the *late fusion* in the paper.
  - Segments of length 10s proved to the best. LOSO employed but dataset other than MAHNOB-HCI.
- „CNN and LSTM-Based Emotion Charting Using Physiological Signals“ [48]
  - EEG preprocessed as 2D images and ECG and GSR used as 1D time series signals.
  - ECG and GSR processed using a 1D CNN-LSTM.
  - The final fusion is done as a Majority Vote Mechanism across ECG, EEG and GSR outputs.

10.2 Machine Learning based Fusion Methods

- „Emotion classification in arousal-valence dimension using discrete affective keywords tagging“ [11]
  - Handcrafted features such as HRV, Breathing rate and amplitude, mean, max, min etc. and other statistical features from first and second difference of the signals to a total of 169 features across multiple modalities.
  - SVM classifier with multiple types kernels evaluated.
  - LOSO not employed.
- „Emotion Recognition Based on Weighted Fusion Strategy of Multichannel Physiological Signals “[40]
  - Handcrafted modality specific features for ECG, GSR, EEG and RA.
  - Weighted fusion strategy by learning weights on validation data.
  - Use LOSO test strategy for testing the system
  - Non-Linear SVM classifier.
- „Emotion Classification in Arousal-Valence Model using MAHNOB-HCI Database. “[38]
  - Handcrafted modality specific features for ECG, GSR, Resp and Temp.
  - One of the few papers that only deals with peripheral signals.
  - Use of LOSO or any other validation strategy not mentioned.
  - SVM classifier with multiple types kernels evaluated.

10.3 Unsupervised Learning in context of physiological signals

- „Attribute-invariant Variational Learning for Emotion Recognition Using Physiology“ [49]
  - Use of MMD instead of KLD in addition to reconstruction loss to enforce structure in latent distribution.
  - Attempt to model representations by considering subject level variance.
  - To achieve this introduce *attribute invariance loss* embedded in MMD loss.
  - Autoencoders not trained on original signals but hand crafted features.
  - Subject-Independent 10-Fold cross validation; LOSO not employed.
  - Authors - ”Our analysis of the extracted physiological LLDs further reveals that “Hjorth” and “ARMPB” from EEG are key factors in bringing insight on how personality affects physiological emotion reaction, and “Creativeness” has a more prominent effect on the cardiovascular measurement”.
- „Unsupervised multi-modal representation learning for affective computing with multi-corpus wearable data“ [50]
  - Training Autoencoders as feature extractors.
  - Use parallel stacked encoders for ECG and EDA.
  - Subject-Wise 0-1 scaling of signals and MSE as reconstruction loss.
  - 10-fold cross validation for classification using best version of autoencoder; LOSO not employed.

10.4 Data

- “ MAHNOB-HCI - A Multimodal Database for Affect Recognition and Implicit Tagging“ [1]
  - Multimodal setup for synchronized recording of face videos, audio signals, eye gaze data, and peripheral/central nervous system physiological signals
  - 30 Subjects
  - Use of self-rating system for valence, arousal and other basic emotions.
  - Videos and Images as stimuli.

Bibliography

[1] Mohammad Soleymani, Jeroen Lichtenauer, Thierry Pun, and Maja Pantic. A multimodal database for affect recognition and implicit tagging. *IEEE Trans. Affect. Comput.*, 3(1):42–55, jan 2012.

[2] Jianhua Tao and Tieniu Tan. Affective computing: A review. In Jianhua Tao, Tieniu Tan, and Rosalind W. Picard, editors, *Affective Computing and Intelligent Interaction*, Seiten 981–995, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[3] Rosalind W. Picard. Affective computing. 1997.

[4] Zhihong Zeng, Jilin Tu, Ming Liu, Thomas S Huang, Brian Pianfetti, Dan Roth, and Stephen Levinson. Audio-visual affect recognition. *IEEE Transactions on multimedia*, 9(2):424–428, 2007.

[5] Pranav Kumar, SL Happy, and Aurobinda Routray. A real-time robust facial expression recognition system using hog features. In *2016 International Conference on Computing, Analytics and Security Trends (CAST)*, Seiten 289–293. IEEE, 2016.

[6] Yuan Zong, Xiaohua Huang, Wenming Zheng, Zhen Cui, and Guoying Zhao. Learning from hierarchical spatiotemporal descriptors for micro-expression recognition. *IEEE Transactions on Multimedia*, 20(11):3160–3172, 2018.

[7] Jian Huang, Ya Li, Jianhua Tao, Zheng Lian, and Jiangyan Yi. End-to-end continuous emotion recognition from video using 3d convlstm networks. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Seiten 6837–6841. IEEE, 2018.

[8] Ling Lo, Hong-Xia Xie, Hong-Han Shuai, and Wen-Huang Cheng. Mer-gcn: Micro expression recognition based on relation modeling with graph convolutional network, 2020.

[9] Dimitrios Kollias and Stefanos Zafeiriou. Exploiting multi-cnn features in CNN-RNN based dimensional emotion recognition on the OMG in-the-wild dataset. *CoRR*, abs/1910.01417, 2019.

[10] Yi Ding, Neethu Robinson, Su Zhang, Qiuhao Zeng, and Cuntai Guan. TSception: Capturing temporal dynamics and spatial asymmetry from EEG for emotion recognition. *IEEE Transactions on Affective Computing*, Seiten 1–1, 2022.

[11] Wiem Mimoun Ben Henia and Zied Lachiri. Emotion classification in arousal-valence dimension using discrete affective keywords tagging. In *2017 International Conference on Engineering MIS (ICEMIS)*, Seiten 1–6, 2017.

[12] Siddharth, Tzyy-Ping Jung, and Terrence J. Sejnowski. Utilizing deep learning towards multi-modal bio-sensing and vision-based affective computing. *CoRR*, abs/1905.07039, 2019.

[13] Yuhao Zhang, Md Zakir Hossain, and Shafin Rahman. Deepvanet: A deep end-to-end network for multi-modal emotion recognition. In Carmelo Ardito, Rosa Lanzilotti, Alessio Malizia, Helen Petrie, Antonio Piccinno, Giuseppe Desolda, and Kori Inkpen, editors, *Human-Computer Interaction – INTERACT 2021*, Seiten 227–237, Cham, 2021. Springer International Publishing.

[14] Enrique Munoz-De-Escalona and José Cañas. Online measuring of available resources. 06 2017.

[15] Stamos Katsigiannis and Naeem Ramzan. DREAMER: A Database for Emotion Recognition through EEG and ECG Signals from Wireless Low-cost Off-the-Shelf Devices. April 2017.

[16] Yarin Gal. *Uncertainty in Deep Learning*. PhD thesis, University of Cambridge, 2016.

[17] H B Barlow. In Geoffrey E. Hinton and Terrence Joseph Sejnowski, editors, *Unsupervised Learning: Foundations of Neural Computation*.

[18] Heli Koskimäki. Avoiding bias in classification accuracy - a case study for activity recognition. 12 2015.

[19] Valorie N Salimpoor, Mitchel Benovoy, Gregory Longo, Jeremy R Cooperstock, and Robert J Zatorre. The rewarding aspects of music listening are related to degree of emotional arousal. *PLoS One*, 4(10):e7487, October 2009.

[20] Hugo D. Critchley. Review: Electrodermal responses: What happens in the brain. *The Neuroscientist*, 8(2):132–142, 2002. PMID: 11954558.

[21] Jonghwa Kim and Elisabeth André. Emotion recognition based on physiological changes in music listening. *IEEE Trans Pattern Anal Mach Intell*, 30(12):2067–2083, December 2008.

[22] Richard A. McFarland. Relationship of skin temperature changes to the emotions accompanying music. *Biofeedback and Self-regulation*, 10(3):255–267, Sep 1985.

[23] Sara Rimm-Kaufman and Jerome Kagan. The psychological significance of changes in skin temperature. *Motivation and Emotion*, 20:63–78, 03 1996.

[24] Taylor W. M. Baker, L. M. The relationship under stress between changes in skin temperature, electrical skin resistance, and pulse rate. *Journal of Experimental Psychology*, 48:361–366, 1954.

[25] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning internal representations by error propagation. In David E. Rumelhart and James L. Mcclelland, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations*, Seiten 318–362. MIT Press, Cambridge, MA, 1986.

[26] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, 1(4):541–551, 12 1989.

[27] Paul J. Werbos. Generalization of backpropagation with application to a recurrent gas market model. *Neural Networks*, 1(4):339–356, 1988.

[28] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, nov 1997.

[29] Jingyuan Wang, Ze Wang, Jianfeng Li, and Junjie Wu. Multilevel wavelet decomposition network for interpretable time series analysis. *CoRR*, abs/1806.08946, 2018.

[30] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting, 2015.

[31] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, Seiten 8024–8035. Curran Associates, Inc., 2019.

[32] Nicki Skafte Detlefsen, Jiri Borovec, Justus Schock, Ananya Harsh Jha, Teddy Koker, Luca Di Liello, Daniel Stancl, Changsheng Quan, Maxim Grechkin, and William Falcon. Torchmetrics - measuring reproducibility in pytorch. *Journal of Open Source Software*, 7(70):4101, 2022.

[33] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[34] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

[35] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors.

SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[36] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[37] Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, and Orion Reblitz-Richardson. Captum: A unified and generic model interpretability library for pytorch, 2020.

[38] Mimoun Wiem and Zied Lachiri. Emotion classification in arousal valence model using mahnob-hci database. *International Journal of Advanced Computer Science and Applications*, 8, 03 2017.

[39] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948.

[40] Wei Wei, Qingxuan Jia, Yongli Feng, and Gang Chen. Emotion recognition based on weighted fusion strategy of multichannel physiological signals. *Comput Intell Neurosci*, 2018:5296523, July 2018.

[41] L. N. Vaserstein. Markov Processes over Denumerable Products of Spaces, Describing Large Systems of Automata. *Probl. Peredachi Inf.*, 5:3, 1969.

[42] Kantorovich LV. Mathematical methods of organizing and planning production. *Management Science. 6*, 6, 1939.

[43] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[44] Yilong Yang, Qingfeng Wu, Ming Qiu, Yingdong Wang, and Xiaowei Chen. Emotion recognition from multi-channel eeg through parallel convolutional recurrent neural network. In *2018 international joint conference on neural networks (IJCNN)*, Seiten 1–7. IEEE, 2018.

[45] Jun-Ho Maeng, Dong-Hyun Kang, and Deok-Hwan Kim. Deep learning method for selecting effective models and feature groups in emotion recognition using an asian multimodal database. *Electronics*, 9(12), 2020.

[46] Yong Zhang, Cheng Cheng, and Yidie Zhang. Multimodal emotion recognition using a hierarchical fusion convolutional neural network. *IEEE Access*, 9:7943–7951, 2021.

[47] Bahareh Nakisa, Mohammad Naim Rastgoo, Andry Rakotonirainy, Frederic Maire, and Vinod Chandran. Automatic emotion recognition using temporal multimodal deep learning. *IEEE Access*, 8:225463–225474, 2020.

[48] Muhammad Najam Dar, Muhammad Usman Akram, Sajid Gul Khawaja, and Amit N. Pujari. Cnn and lstm-based emotion charting using physiological signals. *Sensors*, 20(16), 2020.

[49] Hao-Chun Yang and Chi-Chun Lee. An attribute-invariant variational learning for emotion recognition using physiology. Seiten 1184–1188, 05 2019.

[50] Kyle Ross, Paul Hungler, and Ali Etemad. Unsupervised multi-modal representation learning for affective computing with multi-corpus wearable data. *Journal of Ambient Intelligence and Humanized Computing*, October 2021.

Appendix

Unusual ECG Signals

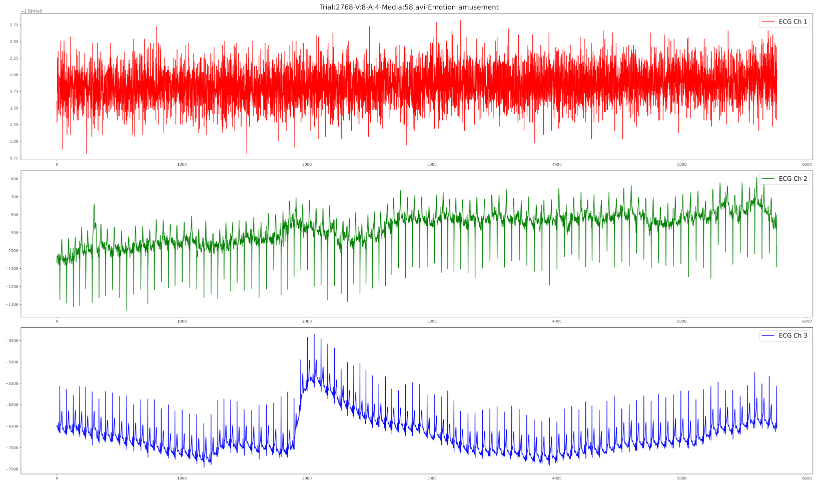


Figure 10.1: ECG data from trial-2768 of MAHNOB-HCI dataset. The trial has a valence and arousal ratings of 8 and 4 respectively. *58.avi*(see section-1.9.3) was used. Notice the unexpected long range of channel in red along with the very high frequency noise not usually present in ECG signals.

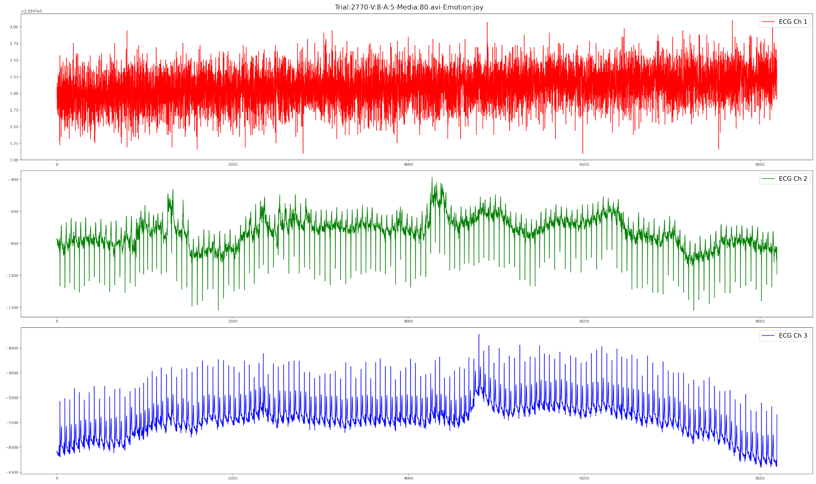


Figure 10.2: ECG data from trial-2770 of MAHNOB-HCI dataset. The trial has a valence and arousal ratings of 8 and 5 respectively. *80.avi*(see section-1.9.3) was used. Notice the unexpected long range of channel in red along with the very high frequency noise not usually present in ECG signals.

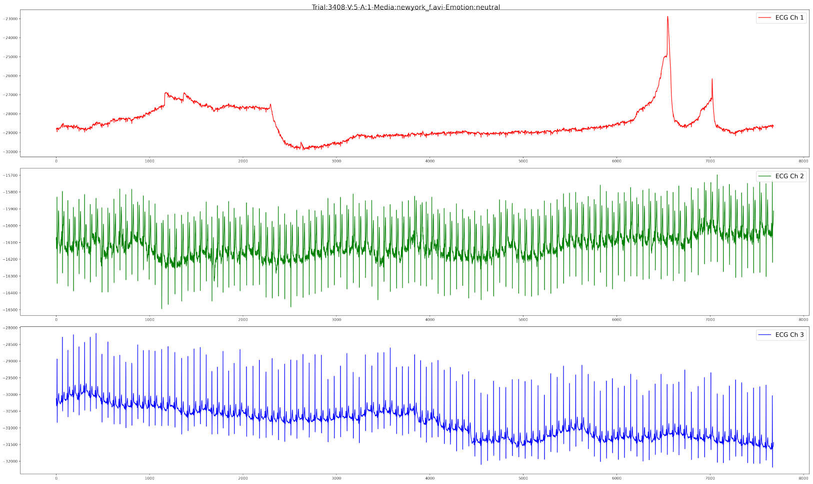


Figure 10.3: ECG data from trial-3408 of MAHNOB-HCI dataset. The trial has a valence and arousal ratings of 5 and 1 respectively. *newyork_f.avi*(see section-1.9.3) was used. Notice the unusual ECG signal from *ECG-Ch 1* which is very different from other two channels. The peak in the signal may also effect the normalization process for this subject's data overall.

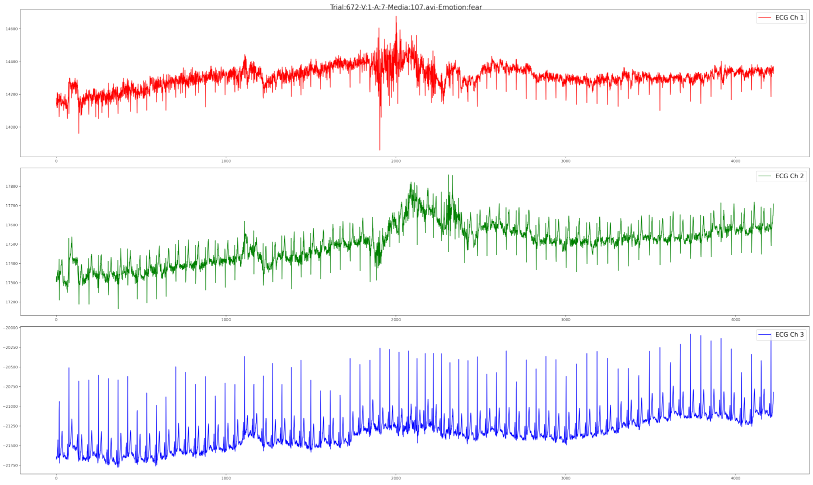


Figure 10.4: ECG data from trial-672 of MAHNOB-HCI dataset. The trial has a valence and arousal ratings of 1 and 7 respectively. *107.avi*(see section-1.9.3) was used.The artifact present mid-way of the trial is again not a proper ECG signal. In-fact the signal looks okay after the middle point. The noisy peak will also effect the normalization process.

Unusual GSR Signals

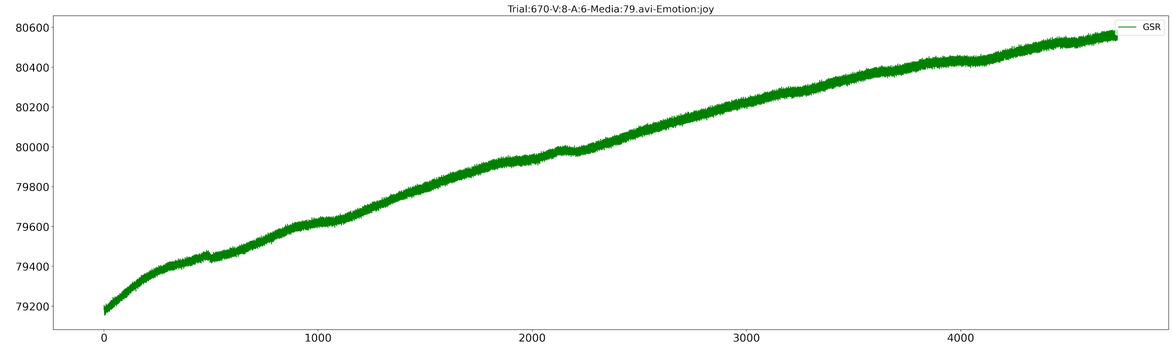


Figure 10.5: GSR data from trial-670 of MAHNOB-HCI dataset. The trial has a valence and arousal ratings of 8 and 6 respectively. *79.avi*(see section-1.9.3) was used.