SAARLAND UNIVERSITY

Faculty of Mathematics and Computer Science
Department of Computer Science
MASTER THESIS

Multimodal transformer for affect analysis in human-virtual agent dyadic interactions

submitted by
Senorita Rodricks
Saarbrücken
November 2024

Advisors:
Chirag Bhuvaneshwara,
German Research Center for Artificial Intelligence, Saarbrücken, Germany
Dr. Dimitra Tsovaltzi,
German Research Center for Artificial Intelligence, Saarbrücken, Germany
Dr. Fabrizio Nunnari,
German Research Center for Artificial Intelligence, Saarbrücken, Germany

Supervisor:
Prof. Dr. Antonio Krüger,
German Research Center for Artificial Intelligence, Saarbrücken, Germany

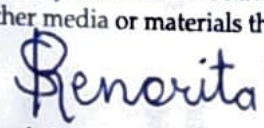
Reviewers:
Prof. Dr. Antonio Krüger,
German Research Center for Artificial Intelligence, Saarbrücken, Germany
Prof. Dr. Patrick Gebhard
German Research Center for Artificial Intelligence, Saarbrücken, Germany

Saarland University
Faculty MI – Mathematics and Computer Science
Department of Computer Science
Campus - Building E1.1
66123 Saarbrücken
Germany

Declarations

Statement in Lieu of an Oath:

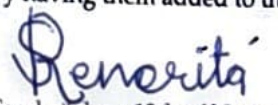
I hereby confirm that I have written this thesis on my own and that I have not used any other media or materials than the ones referred to in this thesis.



Saarbrücken, 12th of November, 2024

Declaration of Consent:

I agree to make both versions of my thesis (with a passing grade) accessible to the public by having them added to the library of the Computer Science Department.



Saarbrücken, 12th of November, 2024

Acknowledgements

I would like to express my sincere gratitude to Prof. Dr. Antonio Krüger for the invaluable opportunity to work on this thesis under his supervision.

I am also deeply thankful to Patrick Gebhard for his insightful feedback and careful review of my work. Special thanks go to my advisor, Chirag Bhuvaneshwara, whose mentorship has been invaluable from the very beginning. I would also like to thank Dmitra Tsovaltzi and Fabrizio Nunnari for the engaging discussions that have helped bring about this thesis.

I am fortunate to have had the support of my lab colleagues, who have made this journey even more enriching. I would like to thank Shailesh, Dawood, Mohammed, and the rest of the ACG group for their camaraderie, insightful conversations, and unwavering support. Their willingness to share knowledge and experiences, whether in formal settings or casual discussions, has been an integral part of my research experience.

Lastly, I extend my heartfelt appreciation to my mom and dad, my loved ones and my best friends Kartik, Anushree, Ena and Gabi, whose unwavering support has been a constant source of strength throughout this journey. Their encouragement, patience, and understanding have allowed me to persevere and bring this work to fruition.

Thank you to everyone who contributed in any way, big or small, to making this thesis possible.

Abstract

Enhancing virtual agents with social skills and human-like intelligence improves human-virtual agent interactions enabling more natural and empathetic responses. To achieve a comprehensive understanding of emotions, we adopt the Pleasure, Arousal and Dominance (PAD) framework, which captures the full emotional spectrum crucial for nuanced affect analysis. Traditional approaches for affect recognition often rely on single-source, sensor-dependent data or often overlook the Dominance dimension, limiting their effectiveness in modeling human-like emotions in virtual agents.

To automate affect analysis in human-virtual agent dyadic interactions, we propose a multimodal machine learning model that leverages non-intrusive audio and video data to make real-time PAD predictions, thereby eliminating the need for wearable sensors. Our model extracts context-aware feature representations from audio and video inputs using Wav2Vec2 and VideoViT models, respectively, and fuses them in a transformer-based fusion model to make predictions. The DEAP and MITHOS datasets are used to train the model as they closely align with our objectives and help capture affective patterns.

The results demonstrate that the fusion model outperforms standalone models, underscoring the advantages of multimodal integration. The fusion model leverages complementary information from both audio and video inputs, allowing a more holistic understanding of user emotions, improving model adaptability to dynamic interaction scenarios.

The inferences from this multimodal model have practical applications in virtual agents, particularly in domains like education and social coaching as done in the MITHOS system. By incorporating PAD-based affect predictions, virtual agents can better interpret and adapt to users' emotional cues, personalizing interactions to enhance engagement, empathy, and rapport. This model paves the way for virtual agents to deliver more contextually relevant and emotionally attuned responses, ultimately enriching user experiences.

Contents

1 Introduction **1**

1.1 Motivation . . . 2

1.1.1 Prediction . . . 2

1.1.2 Measurements . . . 3

1.1.3 Multi-modal approach . . . 3

1.2 Research Goals and outline . . . 4

2 Related Work **5**

2.1 Machine Learning models used for PAD prediction . . . 5

2.1.1 Basic Machine Learning models . . . 6

2.1.2 Deep Learning-based Neural Network Techniques . . . 6

2.2 Fusion-based techniques . . . 7

2.3 Audio Video Multi-modal Fusion models . . . 9

2.4 Comparison to our approach . . . 9

3 Datasets **11**

3.1 DEAP Dataset . . . 12

3.1.1 Analysis . . . 12

3.2 MITHOS Dataset . . . 15

3.2.1 Experiment Setup and Dataset Creation . . . 16

3.2.2 Analysis . . . 17

4 Implementation **21**

4.1 Setup and Environment Configuration . . . 21

4.2 Transfer Learning Overview . . . 22

4.3 Model Architecture Overview . . . 23

4.4 Evaluation . . . 24

4.5 Implementation of model for video feature extraction . . . 26

4.5.1 Training of DEAP dataset . . . 26

4.5.2 Fine-tuning the video model on MITHOS dataset . . . 29

4.6 Implementation of model for Audio feature extraction . . . 31

4.7 Transformer Fusion Model . . . 34

5 Experiments **37**
5.1 Video Model . . . 37
5.1.1 Data Preprocessesing Experiments . . . 37
5.1.2 Model Selection . . . 42
5.1.3 Conclusion . . . 46
5.2 Audio Model . . . 46
5.2.1 Data Preprocessing Experiments . . . 46
5.2.2 Model Selection . . . 46
5.2.3 Conclusion . . . 48
5.3 Fusion Model . . . 48
5.3.1 Data Preprocessesing Experiments . . . 48
5.3.2 Model Selection and Experiments . . . 48
5.3.3 Conclusion . . . 52
5.4 Other General Experiments . . . 52
5.4.1 Model Hyperparameters . . . 52
5.4.2 Dimensionality Configuration: Single vs. Multi-Dimensional Models for PAD Prediction . . . 53

6 Results **55**
6.1 Video Model . . . 55
6.2 Audio Model . . . 60
6.3 Fusion Model . . . 62
6.4 Overall Evaluation . . . 64

7 Conclusion and Discussion **66**
7.1 Conclusion . . . 66
7.2 Discussion . . . 67

Bibliography **69**

Chapter 1
Introduction

Integrating social abilities and human-like intelligence into virtual agents significantly enhances interaction quality, with applications in fields like education and social coaching [9]. By incorporating affect analysis, virtual agents can recognize and respond to users' emotional states, enabling more natural, engaging, and empathetic interactions. Detecting nonverbal cues, such as facial expressions and tone of voice, allows agents to adjust responses to match users' moods and needs, creating a personalized user experience. Research has shown that nonverbal behaviors and personality traits are key factors in building rapport and increasing user satisfaction and acceptance [9]. Therefore, automating affect analysis is essential for virtual agents to deliver responses that genuinely enhance the interaction experience and meet users' emotional needs.

Emotions are psychological and physiological responses to various stimuli and situations that serve as the fundamental elements in affect analysis [14]. The "**P**leasure-**A**rousal-**D**ominance **(PAD)**" model, developed by Albert Mehrabian and James Russel [27], provides a framework that conceptualizes and encapsulates the emotional spectrum, making it ideal for affect analysis in our use case. The Pleasure scale measures the degree of pleasantness or unpleasantness an individual associates with a stimulus, while Arousal gauges the intensity of excitement, energy, or drowsiness experienced [27]. Dominance reflects an individual's perception of control over their environment in a specific situation. Figure 1.1 represents the proposed model and maps the 3 dimensions.

In this thesis, we explored various machine learning models and architectures, carefully evaluating the strengths and weaknesses of each, including state-of-the-art approaches, and constructed a more suitable model that overcomes the limitations of our use case. Our use case is to construct a model that predicts PAD values each ranging from -1 to 1, utilizing non-intrusive multimodal data (i.e., information from multiple sources) to analyze social signals in dyadic interactions between humans and virtual agents. These predictions can be leveraged to generate appropriate responsive behaviors in agents such as robots and virtual assistants.

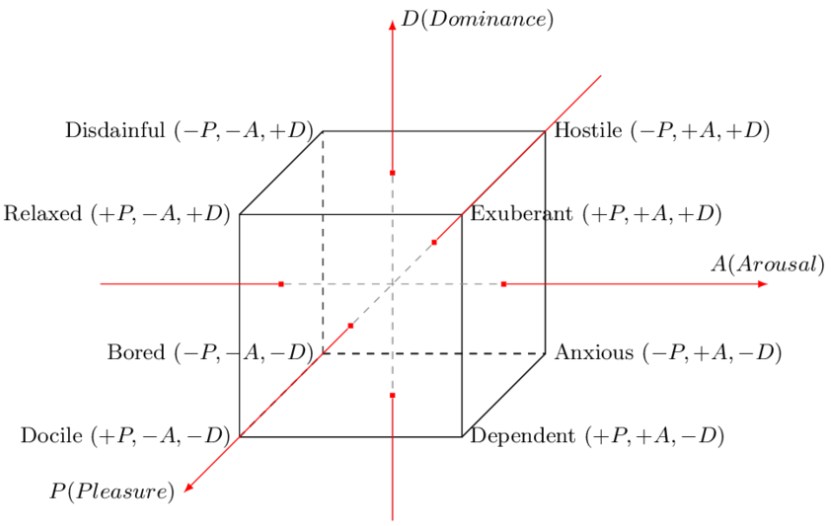


Figure 1.1: The PAD emotional state model developed by Albert Mehrabian and James A. Russell to describe and measure emotional states [27]

1.1 Motivation

1.1.1 Prediction

While Pleasure, Arousal, and Dominance were originally conceptualized as the foundational and independent dimensions of affect analysis, their integration into computational models has been limited, primarily due to the subjective and complex nature of Dominance. Mehrabian's interpretation views Dominance as an individual's sense of control or power in a given situation [27]. This is the definition we consider throughout the thesis. However, other interpretations include how individuals experience and interact with stimuli, how high-potency stimuli can diminish Dominance responses, or as the conative dimension of actions and behavior [3]. As a result, most affective modeling approaches focus solely on Pleasure and Arousal, neglecting Dominance, which provides crucial insights into an individual's control and interaction with stimuli. For example, consider the emotions of **Anger** and **Fear**. Both emotions are characterized by low Pleasure and high Arousal levels, yet they differ significantly in their Dominance values. Anger is associated with high Dominance, which often involves a sense of control or power over the situation. In contrast, Fear corresponds to low Dominance, as individuals typically feel a lack of control or power. This distinction highlights Dominance's critical role in differentiating emotional experiences and its importance in comprehensive affect analysis. [26]

Each dimension of the PAD model captures unique aspects of emotional experience, with even subtle variations revealing important distinctions between states. To address this gap, we incorporate the complete PAD model, based on Mehrabian's original definitions, for a more comprehensive and nuanced understanding of emotions.

	Factor 1 "Pleasure"	Factor 2 "Arousal"	Factor 3 "Dominance"
Unhappy–Happy	*0.914*	0.063	0.148
Annoyed–Pleased	*0.883*	0.068	0.158
Unsatisfied–Satisfied	*0.868*	0.144	0.114
Melancholic–Contented	*0.725*	0.095	0.056
Despairing–Hopeful	*0.858*	0.063	0.078
Bored–Relaxed	*0.580*	0.372	0.234
Relaxed–Stimulated	–0.211	*0.774*	0.052
Calm–Excited	–0.181	*0.793*	0.056
Sluggish–Frenzied	0.268	*0.771*	0.005
Dull–Jittery	–0.211	*0.793*	0.121
Sleepy–Wide awake	–0.046	*0.810*	0.047
Unaroused–Aroused	0.051	*0.827*	0.127
Controlled–Controlling	0.262	0.192	*–0.673*
Influenced–Influential	0.292	0.089	*–0.618*
Cared for–In control	–0.090	0.198	*–0.626*
Awed–Important	0.199	–0.040	*–0.301*
Submissive–Dominant	0.195	0.306	*–0.695*
Guided–Autonomous	0.161	–0.100	*–0.479*
Amount of variance accounted for:	24.6	23.12	12.18

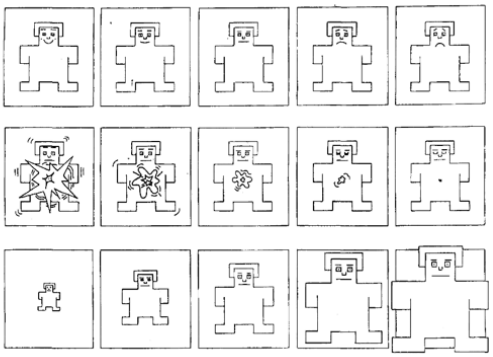


Figure 1.2: (Left) Semantic Differential Scale (SDS)- a questionnaire comprising 6 bipolar adjective pairs that represent each of the PAD dimensions [7]. (Right) Self-Assessment Manikin (SAM) - simple graphical representations (manikins) to assess emotional dimensions [8]

1.1.2 Measurements

Currently, methods such as the Semantic Differential Scale (SDS) [7] and Self-Assessment Manikin (SAM) [8], as shown in Figure 1.2 are used for labeling emotions. SDS is a questionnaire comprising 6 bipolar adjective pairs (like Unhappy-Happy) that represent each of the PAD dimensions to rate their emotional experience [7]. Whereas SAM replaces these verbal descriptors with simple graphical representations (manikins) to assess emotional dimensions [8]. Despite their structured approaches, they heavily rely on retrospective self-reporting, which can introduce inaccuracies because they rely on participants to recall and assess their emotional states after the fact, rather than in real-time. Emotions are dynamic and can change quickly, making it difficult for individuals to accurately recall and quantify how they felt in the moment, introducing memory biases.

We need to develop an automated approach that uses these annotations to train a model capable of real-time PAD predictions. This approach would reduce the need for manual intervention and improve emotional analysis in real-time.

1.1.3 Multi-modal approach

Many existing prediction methods and models rely on data from a single source, such as video [41] or physiological signals like EEG sensors [37], HRV [33], etc., which limits the depth of emotional analysis. Video data offers information about non-verbal cues including facial expressions and body movements [25], while ECG and heart rate data offer insights into the physiological responses associated with emotional experiences [37]. However, relying on a single data source can result in incomplete emotional analysis, as each modality captures only a specific aspect of the emotional spectrum. By leveraging multimodal data—combining video, audio, and physiological signals—we aim to capture a broader range of emotional cues, allowing different modalities to complement one another and overcome the limitations of single-source approaches [23].

Additionally, sensors like the EEG, ECG, etc. require participants to wear specialized equipment, as shown in image 3.1, which adds overhead and makes predictions impractical in their absence. To reduce the overhead associated with external sensors, it

is essential to develop prediction models that do not require participants to wear specialized equipment. By relying on non-invasive modalities such as audio, video, and text—data that can be captured without the need for devices to be placed on the participant—we can achieve effective emotional analysis while maintaining a more practical and participant-friendly setup.

1.2 Research Goals and outline

The primary goal of this research is to develop an affect analysis model that can be integrated into human-virtual agent dyadic interactions to enhance responsiveness and interaction quality. While related work in this field has made strides in affective analysis, existing models face several limitations: they often rely on traditional approaches, that often overlook Dominance [36], use single-source data, depend on intrusive sensor-based modalities [33], and employ architectures lacking scalability [11]. To address these gaps, we propose a machine learning model that predicts Dominance alongside Pleasure and Arousal, enabling a more comprehensive analysis of emotional states. Our model aims to automate emotion prediction using multimodal data—specifically audio and video—without the need for intrusive external sensors. By implementing a transformer-based, late-fusion model with dynamic weighting of modalities based on input relevance, we hope to overcome challenges such as sensitivity to noise and the inability to adapt to variations in data quality, while reducing reliance on manual self-reporting and expensive sensor setups. Furthermore, this research could support systems like ALMA [15], which provide virtual humans with dynamic personality profiles, real-time emotions, and adaptive behaviors to enrich human-agent interactions. The potential of our approach lies in its ability to reduce analysis time and cost, delivering nuanced emotional insights that enhance interaction quality in applications such as education and social coaching, as demonstrated in the MITHOS [10] system.

Document Outline
The next chapter, *Related Work*, reviews existing methods and machine learning models used for affective analysis, highlighting their limitations and establishing the need for our approach. This analysis clarifies the specific gaps our research aims to address. Following this, the *Datasets* chapter introduces the datasets that underpin our machine learning models, providing essential context on the data used to support our proposed approach.

With a clear understanding of the problem and data, we then proceed to the *Implementation* chapter that presents the design of our approach and model architecture for PAD prediction with multimodal data. Subsequently, in the *Experiments* chapter, we detail the series of experiments conducted to refine and optimize our model architecture, documenting the process and rationale behind each decision.

The *Results* section elaborates on the findings from these experiments, presenting the performance outcomes of our model and analyzing its effectiveness in predicting PAD values. Finally, in *Conclusions*, we summarize the key inferences drawn from our results and examine how these insights contribute to the broader field of affective analysis. The thesis concludes with a *Discussion* section, which offers a reflective analysis on our conclusions, considers limitations, and discusses avenues for future work.

Chapter 2
Related Work

In this chapter, we review various methods and machine learning models that have been used to predict Pleasure, Arousal, and Dominance (PAD) values, and assess their suitability for our use case. We focus on models that perform PAD prediction and those that leverage audio-video data for affect analysis, ultimately concluding with a suitable approach for our research. The analysis will also highlight the typical accuracy ranges achieved by different models and how they align with our requirements.

2.1 Machine Learning models used for PAD prediction

This section provides an overview of machine learning models that have been applied to predict PAD values. We begin with simpler models that estimate PAD using basic methods like linear regression and SVMs, and then transition to more advanced techniques such as deep learning. We also explore how different models handle multimodal data inputs and analyze their limitations in capturing emotional cues, especially for Dominance.

In the early stages, estimating Pleasure-Arousal-Dominance (PAD) values was done by calculating the correlation coefficients (r) among the dimensions. For instance, Dominance was found to have a correlation coefficient r=0.71 with Pleasure and r=-0.26 with Arousal, which led to the belief that one dimension could be derived from the other dimensions [26]. However, these correlations were not strong enough to confirm this inference, indicating that the three dimensions are not entirely dependent on one another. Additionally, these basic methods did not account for social cues or the context of the situation, which highlighted the need for more sophisticated computational approaches. In the following subsections, we compare various models used to predict PAD values and analyze their suitability for our use case.

2.1.1 Basic Machine Learning models

Mehrabian employed the Linear Regression model [26], which assumes a linear relationship between input variables and target values. This assumption, however, does not hold in more complex, multimodal scenarios, particularly in high-dimensional settings where linear models struggle to capture intricate, non-linear patterns. The Naïve Bayes method, adapted by Romeo et al. [36], overcomes the linear assumption but operates under the assumption that the input features are completely independent. For example, it assumes no correlation between facial expressions and body movements, which is unrealistic in our multimodal scenario where features interact in complex ways. Additionally, Naïve Bayes is not well-suited for processing unstructured data types such as images, audio, or video, prompting the exploration of more robust methods.

Support Vector Machines (SVMs) are better at capturing non-linear patterns, making them powerful tools in machine learning. Romeo et al. [36] used SVMs to predict Pleasure and Arousal based on physiological signals like Galvanic Skin Response (GSR), Blood Volume Pressure (BVP), Respiration (RESP), Electromyography (EMG), Inter-beat Interval (IBI) for heart rate, and Skin Temperature and achieved a balanced measure of performance by considering both precision and recall for continuous values. Kim et al. [17] also predicted PAD values using a two-class (High and Low values) classification method using the SVM model. Petrescu et al. [33] applied SVMs for fear emotion classification based on physiological data, including electrodermal activity (EDA) and heart rate variability (HRV), where fear was associated with low valence, high arousal, and low dominance. Despite their strengths, SVMs suffer from the "curse of dimensionality," which hampers performance as the feature space expands [33]. Moreover, class imbalance is a common issue in SVMs, where certain labels may appear more frequently than others, causing poor model performance [4].

The Weighted K-nearest neighbor (kNN) classifier, a variant of kNN, was also used by Kim et al. [17] for PAD prediction based on poster designs. Although kNN models perform well for Dominance prediction, they may work better only in low-dimensional datasets and struggle with complex, noisy data. Kim et al. also applied Decision Trees and Random Forests for emotion classification [17], but Decision Trees tend to be unstable when exposed to noise, while Random Forests, though more stable, may not explore all possible feature combinations in high-dimensional datasets, risking overfitting.

While PAD dimensions are three independent dimensionalities, our use of inputs in the form of multi-modalities suggests the existence of interdependencies that warrant investigation for advanced methods that incorporate various dimensionalities, patterns, etc. For example, an angry participant may reveal patterns in their facial expressions and voice modulation [12]. Additionally, the context resulting in a certain response is not captured by these basic machine learning models. As the above models don't capture these features in the data, our investigation delved into mature deep-learning algorithms.

2.1.2 Deep Learning-based Neural Network Techniques

In this section, we explore deep learning models like Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) networks, and Transformer models, which have been applied to capture more complex patterns in multimodal data. These models outperform traditional methods in tasks requiring high-dimensional, non-linear data processing.

Convolutional Neural Networks (CNNs) excel in handling intricate aspects of input

modalities, such as semantics and their interrelationships like a frown from video data and jitter in the voice from audio data [12]. Nandini et al. [30] and Li et al. [22] use multi-channel EEG signals to predict emotional states using the CNN model. Similarly, Petrescu et al. [33] use a Photoplethysmogram (PPG) to detect blood volume changes and Heart Rate Variability (HRV) signals to detect variability in the heart rate to predict Pleasure and Arousal related features with the CNN model. Even though CNNs capture these local patterns, they fail to make inferences about the context that brings about these patterns i.e. capture temporal information [22]. The Long Short-Term Memory Networks (LSTMs) have the ability to capture long-range dependencies representing relationships among elements in a sequence, even when separated by a substantial number of time steps i.e. it captures the context of a given situation., compensating for the shortcomings of the CNN model. LSTMs yield impressive results in capturing temporal information; however, they may not perform well in capturing local patterns within the data and therefore complement the CNN models [1]. To address the limitations mentioned earlier, combining the CNN and LSTM models was a natural progression. This hybrid approach enables us to simultaneously capture local features within a time frame and long-range dependencies among multiple time frames [13] capturing the spatial and temporal relationships crucial for our study. However, when we consider multiple modalities, having a CNN-LSTM model for each modality could result in a complex architecture with a very high number of parameters and lack the mechanism to prioritize the most relevant parts of the input effectively.

Transformer models introduced by Ashish Vaswani, et al. [40], have a built-in Attention mechanism that enables the network to establish relationships between each unit of the input with every other unit, making them suitable for inter and intra-modal fusion tasks, where establishing correspondence across different modalities is essential. Attention mechanisms allow the model to weigh the importance of different parts of the input sequence, enabling it to focus on the most relevant information while processing inputs like video frames or audio signals [40]. Meng and Liu [28] estimate Pleasure and Arousal using a transformer-based encoder that captures the temporal and context information from videos in the Wild and achieves better results than prior models for Pleasure and Arousal. Plisiecki et al. [34] presents a transformer-based neural network model designed to extrapolate emotional norms (Pleasure, Arousal, Dominance) for words (embeddings) in multiple languages, including English, Polish, Dutch, German, French, and Spanish. This is achieved by using contextual word representations derived from transformers and pre-trained models fine-tuned for emotion recognition tasks. The Wav2Vec2 Model created by Hugging Face predicts the PAD values based on audio signals on MSP-Podcast [42].

We see that Transformers models have now shown significant success in various domains like videos [28], text data[34], audio signals[42], and similar use cases and hence can be adapted for tasks handling input data from different modalities. However, the above-mentioned models still predict Pleasure, Arousal (and Dominance) values only with single modalities and therefore, in the next sections, we elaborate on leveraging models that use multiple sources of data to process and predict PAD values.

2.2 Fusion-based techniques

In multimodal data processing, the fusion of different modalities is crucial for enhancing the predictive power of models, especially in complex tasks like PAD prediction as each modality captures different aspects, and combining them allows us to gain a more holistic

understanding. In this section, we explore various fusion techniques, including early fusion, late fusion, and model-based fusion, and analyze how they integrate multimodal data to improve PAD prediction accuracy. By understanding the strengths and limitations of these approaches, we can determine the most effective strategy for leveraging multiple sources of information in our research.

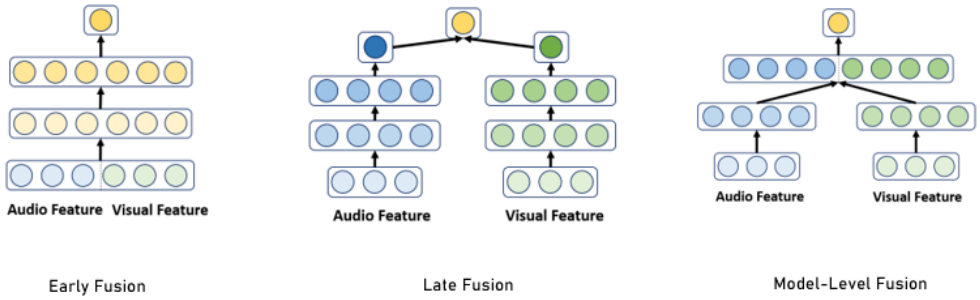


Figure 2.1: Types of Multimodal Fusion [11]: Early Fusion combines audio and visual features at the input level; Late Fusion processes features separately before merging their outputs; Model-Level Fusion integrates features processed by separate models, and processes these features with a different model

Early fusion, also known as feature-level fusion, is an approach in multimodal data processing where information from different modalities is combined at an early stage by merging the features into a single representation, which is then used as input for subsequent processing or analysis [11], as shown in Fig. 2.1. However, as our input modalities are heterogeneous, they may fail to adapt to the complexities that come with it. Scaling the model to incorporate more modalities can add computational and architectural overheads for integration in the future as the complexity of managing and processing fused features at the initial stage becomes challenging.

Late Fusion is particularly apt for our scenario due to the presence of heterogeneous modalities, including audio and video. Heterogeneous data encompasses variations in data types, lengths, formats, structures, content, and imbalances in feature sizes [39], as shown in Fig. 2.1. This technique allows us to independently process these modalities based on their inputs, extract relevant features, and subsequently combine them using another model to address the regression problem. After undergoing feature extraction, these features are fed into their respective models, which are then passed to an independent data fusion model responsible for computing the final output across all inputs as shown in Fig 2.1.

The late fusion model in [32] employs one CNN model to process each input source and passes the outputs to a Two-Stream Inflated 3D ConvNet (I3D) data fusion network for automated human emotion analysis, surpassing the performance of any uni-modal or single model counterparts. These counterparts have fixed architectures, making it challenging to adapt to the specific requirements of each modality making the fusion models less susceptible to noise or errors in individual modalities and compensating for each other's limitations and redundancies [39]. However, this late-fusion technique assigns a predefined weighted sum to different modalities which determines which modality has how much impact. On the contrary, in this thesis, we aim to determine the impact of each modality dynamically rather than assigning predefined weights, allowing the relevance of each modality to shift based on contextual cues and enhancing the

model's robustness and resilience to noise.

Model-based fusion is a special kind of late-fusion technique that accepts feature representations from various modalities preprocessed by their respective models and fuses them to perform a specific task, as shown in Fig. 2.1. R Gnana Praveen uses this technique in a cross-attention module that leverages both inter and intra-modal relationships, thereby significantly improving the performance of the system [35]. This helps in detecting the impact of each modality in a particular situation rather than already pre-defining them. These qualities make the model-based transformer fusion technique a good choice for our scenario.

Now that we have seen the different fusion techniques, we will delve deeper into the different audio-video models that have been designed for prediction.

2.3 Audio Video Multi-modal Fusion models

While previous machine learning models have achieved promising results by leveraging the strengths of various modalities, none of them jointly utilize both audio and video for PAD prediction. These models focus on single modalities or rely on physiological sensors like EEG, ECG, etc. Note that in this thesis, we consider only the audio and video modalities as our thesis focuses on non-intrusive sources that do not require external devices to be worn by the participant. This section focuses on a few models that use audio and video modalities for general prediction.

The Multimodal Audio-Image and Video Action Recognition (MAiVAR) developed by Shaikh et al. [38] is an audio-image and video fusion-based deep learning framework designed for action recognition like playing the guitar, bowling, chopping wood, etc. This uses input modalities like RGB video frames from videos and audio samples. The video features are extracted with a BNInception backbone and are grouped into 25-frame segments and the audio features are extracted with an IRV2 (Inception-ResNet v2) model, pre-trained on ImageNet, and converted into six distinct image-based representations. These extracted features are then passed through a Multimodal Fusion Network (MFN) designed for action recognition [38]. Along similar lines, Hessel R. Bosma developed a General Audio-Visual Transformer (G-AVT), an audio-visual framework that leverages cross-modal attention in self-supervised transformer-based models for learning tasks [6]. The audio and video inputs are passed through 2 different ResNet18-based convolutional feature extractors, which produce a set of fixed-length vectors to then pass to the Transformer module. Similarly, Junwen Xiong et al. [43] introduce DiffSal, a joint Audio and Video learning model for Diffusion Saliency Prediction.

These models are not designed for PAD prediction but can be essentially modified and trained on relevant data for it.

2.4 Comparison to our approach

In contrast to previous work, which predominantly focuses on predicting Pleasure and Arousal [33] [28], leaving Dominance underexplored, our approach predicts all three PAD values—Pleasure, Arousal, and Dominance—providing a more comprehensive affect analysis. This provides a nuanced, continuous measure of emotional states, capturing subtle variations in intensity and control that distinct emotion categories cannot. Many existing studies focus on tasks like emotion classification [17], fear detection [33], or

general emotion prediction [22][34], rather than directly estimating PAD values. Our approach, however, specifically targets all three PAD dimensions, ensuring a complete representation of emotional states.

Additionally, a significant portion of prior research relies on intrusive modalities such as EEG [37] and HRV [33], which provide strong signals leading to high accuracies, but are impractical for everyday applications and can interfere with real-world evaluations. Our solution, by contrast, employs non-intrusive audio and video modalities, making it more feasible for real-time affect analysis without the need for wearable sensors.

Furthermore, many architectures used in previous studies have notable limitations. For instance, early fusion techniques fail to handle heterogeneous modalities effectively, leading to the loss of important modality-specific information when data sources differ in structure and scale. Single-source models also lack the robustness necessary for complex multimodal affect analysis. Our approach, which applies model-based late fusion techniques, overcomes these limitations by independently processing each modality, extracting the features by combining them in a way that preserves critical information, and processing this information that ensures a more accurate and adaptable solution for PAD prediction. The architecture consists of Video ViT [2] and Wav2Vec2 [42] models for feature extraction from video and audio, respectively. These features are then passed to a transformer-based fusion model that uses attention mechanisms to dynamically assign weights to each modality based on the input context. This adaptive weighting allows the model to account for the importance of different modalities, improving the robustness and accuracy of the predictions instead of assigning weights to modalities beforehand [11].

Lastly, existing approaches lack a unified metric for evaluating PAD prediction accuracy. When PAD values are predicted, the outputs are not always standardized between -1 and 1, and the accuracy metrics vary widely, often involving the nature of the outputs, percentage accuracies based on correct predictions [33] [22], class-based classifications (High or Low classes) [17] or correlation factors [28] without consistent benchmarking. In our approach, we aim for rigorous evaluation using Mean Absolute Error and Correlation factors offering a more reliable and interpretable assessment. Mean Absolute Error (MAE) measures the average magnitude of errors between predicted and actual values, with lower values indicating better model performance, so it should be minimized for optimal accuracy. Pearson's Correlation Coefficient (PCC) measures the strength and direction of the linear relationship between predicted and actual values, ranging from -1 to 1, with higher values closer to 1 indicating better model performance, so it should be maximized.

Chapter 3
Datasets

For this research, it was essential to identify datasets that meet domain-specific requirements, specifically the inclusion of Pleasure, Arousal, and Dominance labels across multiple modalities. We examined several widely used datasets, including AffectNet [29], EmoReact [31], AFEW-VA [19], SEWA [20], DEAP [18], and MITHOS [5] [10]. However, as shown in Table 3.1, most of these datasets do not fulfill all our criteria. Notably, AffectNet, EmoReact, AFEW-VA, and SEWA lack labeled Dominance values, an essential dimension for our study. Furthermore, some of these datasets either lack accurate annotations or include alternative affective labels that do not align with our PAD model requirements.

Given these limitations, we selected the DEAP and MITHOS datasets, as they offer the complete PAD labels alongside multimodal data and align well with our research objectives. The following sections provide an in-depth overview of these datasets, including initial analyses relevant to our study.

Dataset Name	**Pleasure**	**Arousal**	**Dominance**	**Other**	**Drawback details**
AffectNet	Yes	Yes	No	No	Data is not accurately labeled
EmoReact	No	Yes	No	Yes	Basic Emotion labels
AFEW-VA	Yes	Yes	No	No	Valence and Arousal labels only
SEMAINE	Yes	Yes	No	Yes	Power label is present. Definition not inline with Dominance
SEWA	Yes	Yes	No	Yes	Liking, Agreement labels are present. Sensitive to cultural differences
DEAP	Yes	Yes	Yes	Yes	Mentioned in section 3.1
MITHOS	Yes	Yes	Yes	Yes	Mentioned in section 3.2

Table 3.1: Comparison of different datasets and their labels

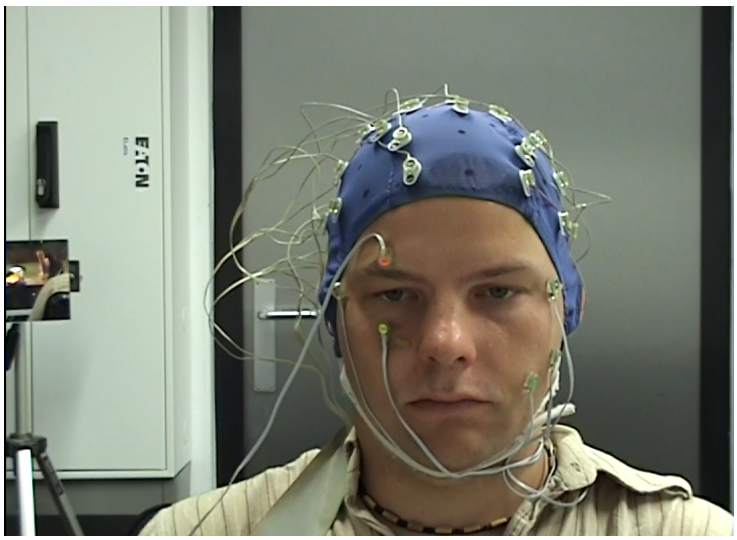


Figure 3.1: Participant wearing an EEG sensor having his face video recorded during an experiment in the DEAP dataset [18]

In the next sections, we elaborate on the DEAP and MITHOS datasets and their initial analysis.

3.1 DEAP Dataset

The DEAP (A Database for Emotion Analysis using Physiological Signals) dataset [18] was designed and developed by Queen Mary University, London, to study human emotions and affective states by capturing physiological and psychophysiological responses consisting of labeled PAD values. The data collection involved 32 healthy participants, aged between 19 and 37, with an equal gender distribution. Participants watched 40 one-minute video clips chosen for their emotional content, selected using a semi-automated process involving affective tags. Participants then rated their emotions using self-assessment manikins (SAM) for Pleasure, Arousal, and Dominance (PAD). The ratings are provided on a continuous scale from 1 to 9, with each dimension capturing different aspects of emotional responses. The physiological signals, including 32-channel EEG, ECG, GSR, and other responses, were recorded using a Biosemi ActiveTwo system. However, only 22 participants had their frontal face videos recorded, as shown in Figure 3.1, that captured their facial expressions while they watched the videos. For our experiments, we only use these frontal face videos.

3.1.1 Analysis

The Queen Mary University provided the DEAP dataset in the form of preprocessed EEG signals and raw video recordings for each participant and their 40 trials. The EEG signals and corresponding labels were saved in DAT files for each participant, while the raw videos captured during each experiment session were made available separately. Additionally, the dataset included metadata for each experimental trial, containing information such as participant questionnaires, ratings, and the list of video clips used

for the experiments. For our experiments, as we focus on the video data, we utilized only the raw video files as input and extracted the output labels for Pleasure, Arousal, and Dominance from the participants_ratings file. This allowed us to align each video segment with the corresponding PAD values for our analysis.

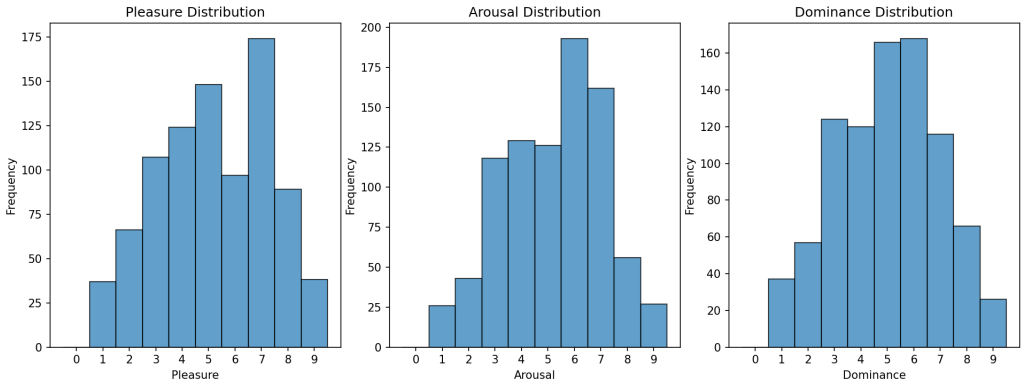


Figure 3.2: Distributions of Pleasure, Arousal, and Dominance values in the DEAP dataset. The histograms show the frequency of each value for the three dimensions, highlighting the range and central tendencies in the dataset.

Dimension	Mean	Median	Mode	Std Dev
Pleasure	5.2180	5.04	1.0	2.0926
Arousal	5.2389	5.49	9.0	1.8786
Dominance	5.0331	5.04	5.04	1.9591

Table 3.2: Summary Statistics for Pleasure, Arousal, and Dominance in the DEAP dataset

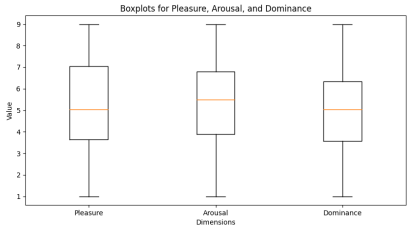


Figure 3.3: Boxplots for Pleasure, Arousal, and Dominance values in the DEAP dataset. The boxplots show the distribution, median (orange line), and range of values for each dimension.

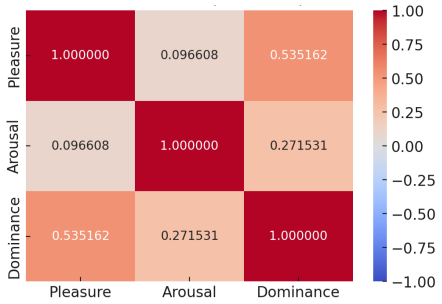


Figure 3.4: Confusion Matrix that shows correlation between the PAD dimensions in the DEAP dataset

The DEAP dataset's Pleasure, Arousal, and Dominance values were analyzed to understand their distribution in Fig. 3.2, central tendencies in Fig. 3.3, and correlations in Fig. 3.4. The analysis helps to evaluate the balance of the dataset and the relationships between these affective dimensions, providing insights into their variance and interaction.

As shown in Fig. 3.2, the histograms illustrate that the distributions of Pleasure, Arousal, and Dominance are generally normal, with values ranging from 1 to 9. The boxplots in

Figure 3.3 indicate that the median values for all three dimensions are centered around 5, suggesting that the data is balanced with a slight variation across the different dimensions. This distribution ensures that the dataset covers a broad range of affective states, which is essential for robust model training.

The summary statistics presented in Table 3.2 provide additional details. The mean values for Pleasure, Arousal, and Dominance are 5.218, 5.239, and 5.033, respectively, indicating that most ratings cluster around the midpoint of the scale. The standard deviations—2.0926 for Pleasure, 1.8786 for Arousal, and 1.9591 for Dominance—demonstrate that Arousal has the least variability, while Pleasure shows the most variation in ratings. Modal values suggest that participants tended to give higher ratings for Arousal, whereas Pleasure and Dominance ratings are often concentrated around lower scores.

The correlation matrix in Fig. 3.4 reveals the relationships among the three dimensions. A correlation matrix displays the correlation coefficients between pairs of variables, with values ranging from -1 to 1, where -1 indicates a perfect negative correlation, 0 indicates no correlation, and 1 indicates a perfect positive correlation. A moderate positive correlation (0.5352) between Pleasure and Dominance suggests that higher levels of Pleasure tend to accompany higher levels of Dominance. In contrast, Arousal shows weaker correlations with both Pleasure (0.0966) and Dominance (0.2715), indicating that changes in Arousal levels are more independent from the other two dimensions, which is also proved in the next chapters.

The dataset was further analyzed based on class-based splits, with DEAP values ranging from 1 to 4 categorized as the "Low" class and values from 5 to 9 as the "High" class. For Pleasure, 54.55% of the samples fall into the "High" class, while 45.45% fall into the "Low" class. Arousal has a greater proportion of samples in the "High" class (58.98%) compared to the "Low" class (41.02%). Dominance is similarly skewed towards the "High" class (56.70%), with 43.30% of samples classified as "Low". This analysis indicates that the DEAP dataset is well-distributed across the three affective dimensions, providing a suitable foundation for training models to predict Pleasure, Arousal, and Dominance. The moderate correlation between Pleasure and Dominance emphasizes the need to account for their relationship during prediction, while the weak correlation with Arousal supports treating it as an independent target.

Limitations

The DEAP dataset proves to be valuable for our research due to its domain-specific labels, specifically focusing on all 3 PAD values with multiple modalities. However, it exhibits a small sample size, comprising only 22 participants with video data. The dataset assigns a single PAD value to each 1-minute video of each individual, which does not account for the variations that can occur in different scenarios within a single video. Upon closer examination of the data quality, we observe the significant variance in labels across participants within a single video explained in Fig 3.5. Eg. Dominance values range from 1 to 9 for nearly every video across participants. Such variability presents challenges in achieving stable and reliable data, which is a crucial requirement for our research. Each of the 22 participants in the dataset originally had 40 video recordings, but participants 3, 5, and 14 had one video missing, and participant 11 had three videos missing. As there was no way to identify which specific videos were missing from their sets, we decided to exclude these participants from the analysis to avoid potential mistraining of the model due to incomplete data.

Despite the problems, we use this dataset to pre-train our model as it helps us study humans' responses through multi-modalities by providing 720 minutes of data to pre-

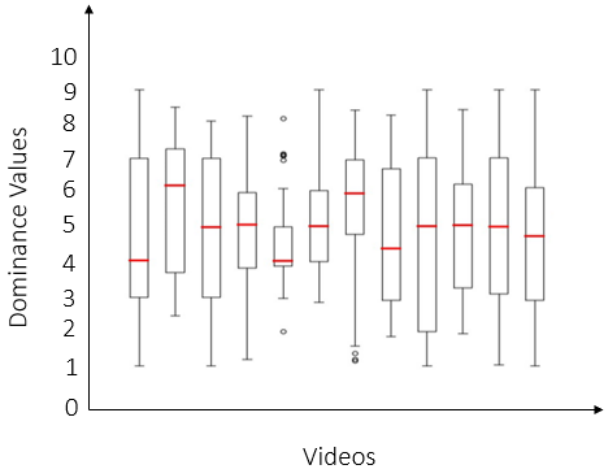


Figure 3.5: Variance of labels for each video sample across participants in DEAP data

train our model.

3.2 MITHOS Dataset

The MITHOS is a system developed by the Affective Computing Group in DFKI, aimed at training teachers' conflict resolution skills through realistic situative learning opportunities during classroom conflicts[5] [10]. The dataset is designed to analyze and train conflict resolution skills in teachers while receiving feedback from socially interactive student agents (SIA) through a virtual controlled environment. The participants include pre-service and in-service teachers who interact with the virtual classroom, facing scenarios like inattentive or confrontative behavior from virtual students, using the Wizard-of-Oz (WoZ) system. The WoZ setup involves a human-wizard, a study master, who assesses the teacher's behavior and models the virtual student's responses in real-time. The data is collected using multiple devices, including web cameras to capture videos, microphones to capture audio signals and physiological signals like Heart rate sensors to monitor physiological responses to stress.

The MITHOS dataset differs significantly from the DEAP dataset in both setup and interaction contexts. While MITHOS involves a realistic situative learning environment where teachers interact with a virtual classroom of students, including a primary student and two additional virtual "audience" students, DEAP lacks any interaction partner or audience. In MITHOS, participants face dynamic scenarios with virtual agents responding to their actions in real-time, simulating social and emotional challenges that can arise in classroom settings. In contrast, the DEAP dataset focuses on passive responses to preselected video clips, with no active engagement or situational feedback from interaction partners. Additionally, MITHOS uses a Wizard-of-Oz (WoZ) setup to model real-time student responses, creating a complex, socially interactive environment, whereas DEAP is limited to capturing physiological and emotional responses without interactive or adaptive elements. These differences make MITHOS particularly well-suited for analyzing emotions in active, social interaction scenarios, as opposed to the more controlled,

non-interactive setting of DEAP.

3.2.1 Experiment Setup and Dataset Creation

The experiments are conducted in a mixed-reality environment that allows teachers to interact with virtual student agents. The study involves multiple training stages, such as situative training, avatar-replay, agent-feedback, and expert-feedback stages. During these interactions, a Wizard of Oz (WoZ) setup is initially employed to guide the responses of virtual agents based on the teachers' behavior, allowing for detailed and tailored interactions. Data is collected through audio-visual sensors, including microphones, cameras, and facial expression trackers, capturing the participants' verbal and non-verbal behaviors. The collected data is labeled based on the interactions between the teachers and the virtual agents, focusing on their emotional responses and conflict resolution styles. Specific emotional states, such as Pleasure, Arousal, and Dominance (PAD), are annotated using self-reports of the participants that are filled in by the participants after the MITHOS interaction. Additionally, social norms and behavior evaluation (e.g., empathy levels, and conflict resolution strategies) are incorporated into the labeling process to reflect the nature of the interactions and the socio-emotional dynamics observed during training. These PAD values are recorded on a scale of 1-9.

The MITHOS dataset captures the interactions between teachers and virtual student agents in a controlled environment using a Wizard-of-Oz (WoZ) setup, as shown in Fig. 3.6. In this process, a human "wizard" who is a study master observes the teacher's behavior and accordingly selects a simplified questionnaire about the affective state of the teacher which is then converted into discrete affective state information by a computational model setup in VSM for the virtual student agents. This is stored in a "State" file and includes data such as the emotional and social states of the interaction, derived from real-time observations and analysis. The responses are then recorded in a "Command" file and sent to the TriCat system, which controls the virtual environment. The commands trigger specific animations and actions for the virtual student agents, such as gestures, facial expressions, or verbal responses, and these are recorded in "action" logs. The actions of the virtual students elicit reactions from the teachers, such as changes in their posture, facial expressions, or tone of voice, which are captured using the Social Signal Interpretation (SSI) framework. The SSI logs, livestream and recordings store multimodal data streams including audio, video, and physiological responses, providing detailed insights into the teacher's behavior during the interaction. The study master continuously reassesses the teacher's responses based on these logs, updating the affective information in the "State" file and iteratively selecting new commands. This iterative process allows for the dynamic adjustment of the virtual student behaviors in real-time, creating a rich, multimodal dataset. This dataset serves as a valuable resource for analyzing teacher responses and training them in effective conflict resolution and socio-emotional regulation skills. The system architecture overview of this process is illustrated in Fig 3.6 of the paper, highlighting the flow of information between the study master, virtual agents, and logging systems.

To analyze the interactions between teachers and virtual student agents, we define a "cue" as the moment when the study master marks a specific reaction observed in the teacher. Each experiment session consists of six cues, enabling us to segment each session into six distinct parts to examine how teachers respond to various actions triggered by the study master. To ensure accurate analysis, especially given the multimodal nature of the data, precise timestamp alignment across modalities is crucial.

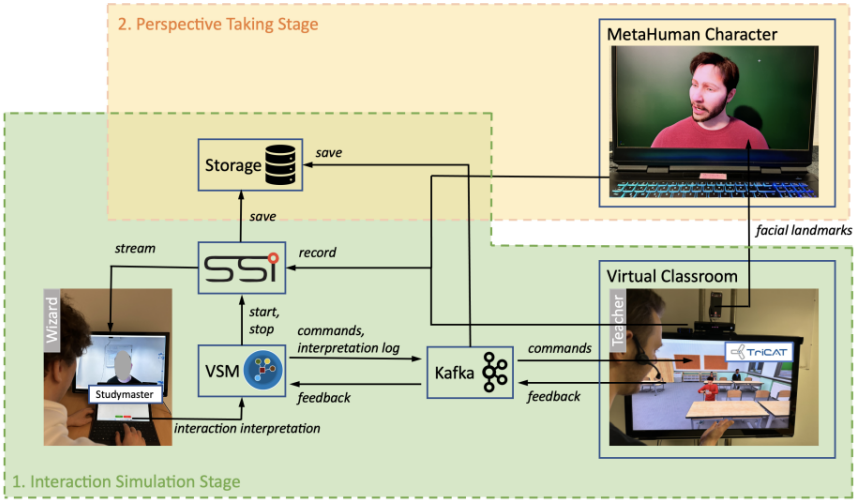


Figure 3.6: System architecture overview of MITHOS Wizard of Oz (WoZ) setup that depicts the setup spread over two rooms and the different system components that the participant and wizard, respectively interact with. [10]

To achieve this, when a reaction is noticed by the study master, the affect information is recorded. However, we also consider the time leading up to this cue, capturing the context of the teacher's reaction. Specifically, we take the start timestamp as the end timestamp of the previous action recorded in the SSI logs right before the cue. This ensures that the period of the teacher's reaction is fully included. The study master's time for evaluation is also taken into account. The end timestamp is derived from the "State" file logs, marking the conclusion of the interaction as determined by the study master. This approach allows for a comprehensive alignment of data segments, ensuring that we accurately capture and analyze the teacher's responses to align across different modalities, which is consistent with best practices for synchronizing multimodal datasets.

We then create an Excel file that records key information for each interaction, including the participant number, cue number, cue timestamp, Pleasure, Arousal, and Dominance (PAD) values, as well as the start and end timestamps computed for each cue. This file serves as a reference point for segmenting the audio and video data according to the specified timestamps, enabling us to accurately analyze and predict the PAD values associated with each reaction segment.

3.2.2 Analysis

The histograms in Fig. 3.7 show the frequency distribution for Pleasure, Arousal, and Dominance. Pleasure is relatively evenly distributed, with a slight concentration around the middle values (4-6). Arousal values tend to be lower, with most values concentrated between 1 and 3, indicating that lower arousal states are more frequent. Dominance shows a peak around the middle range (values 3-5), suggesting a balanced perception of control during interactions.

The boxplots in Fig. 3.8, Fig. 3.9 and Fig. 3.10, represent the distribution of PAD values across six different timestamp cues. For Pleasure, there is a gradual increase in values

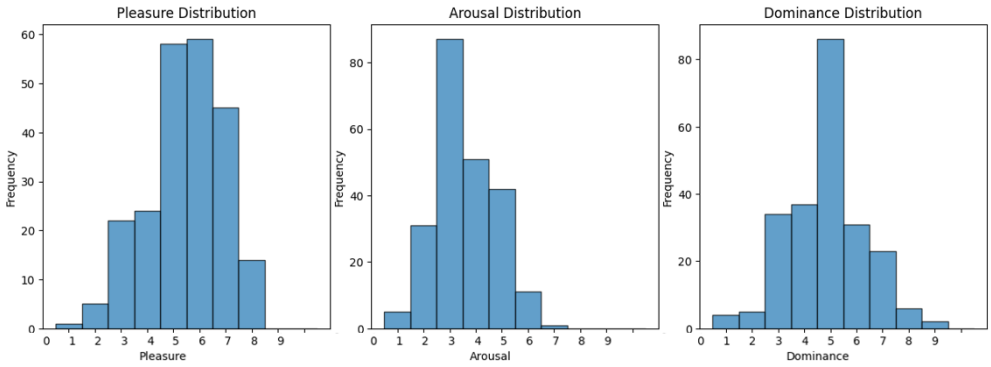


Figure 3.7: Distributions of Pleasure, Arousal, and Dominance values in the MITHOS dataset. The histograms show the frequency of each value for the three dimensions, highlighting the range and central tendencies in the dataset.

as the session progresses, indicating a shift toward a more positive affect. Arousal remains relatively consistent, though it shows some variation in the middle timestamps. Dominance increases over time, suggesting that teachers perceive a growing sense of control throughout the interaction sessions.

The mean, median, mode, and standard deviation for each dimension are presented in Table 3.3. Pleasure has a mean of 4.4561 and a median of 5.0, indicating a tendency towards positive affect. Arousal has a lower mean of 2.5746, reflecting generally lower arousal states. Dominance has a mean of 3.8553, suggesting a moderate perception of control. The standard deviations indicate that the variability in Pleasure and Dominance is higher than that in Arousal.

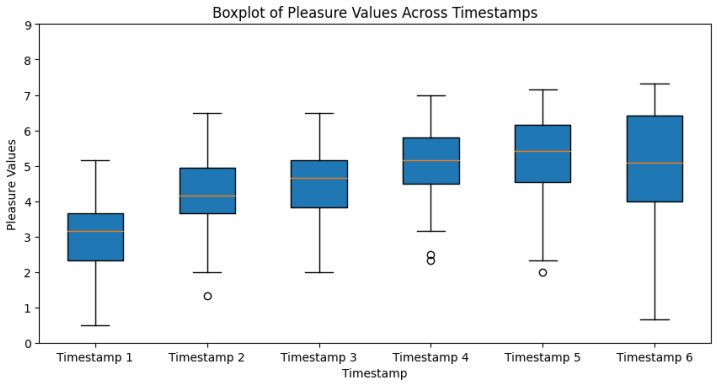


Figure 3.8: Boxplot for Pleasure values in the MITHOS dataset

Dimension	Mean	Median	Mode	Std Dev
Pleasure	4.4561	5.0	5.0	1.4701
Arousal	2.5746	2.0	2.0	1.1681
Dominance	3.8553	4.0	4.0	1.4635

Table 3.3: Summary Statistics of Pleasure, Arousal, and Dominance in the MITHOS dataset

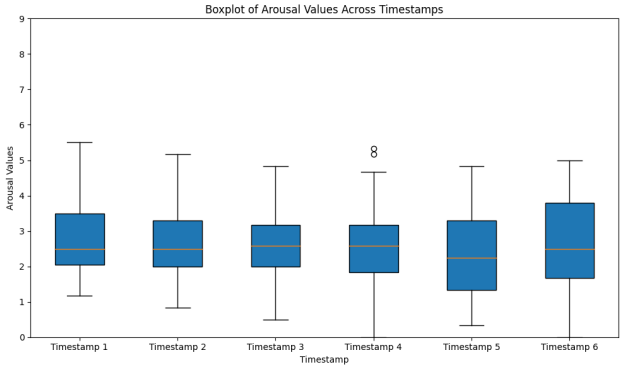


Figure 3.9: Boxplot for Arousal values in the MITHOS dataset

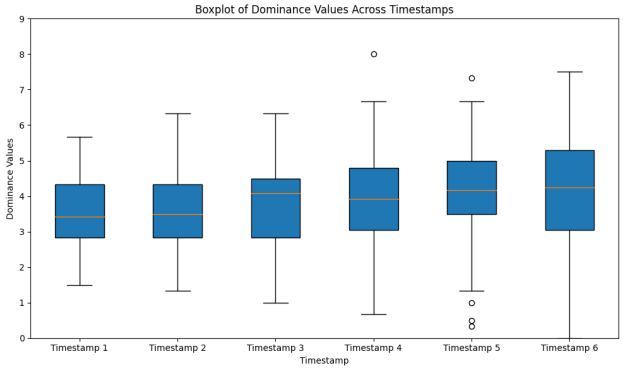


Figure 3.10: Boxplot for Dominance values in the MITHOS dataset

The correlation matrix in Fig. 3.11 highlights the relationships between the three dimensions. Pleasure and Dominance exhibit a positive correlation (0.45), indicating that as positive affect increases, a sense of control also tends to rise. Arousal and Pleasure show a negative correlation (-0.28), suggesting that higher arousal is often associated with less positive affect. Arousal and Dominance have a weak negative correlation (-0.09), indicating minimal interaction between these two dimensions.

The MITHOS dataset was similarly analyzed based on class-based splits, with values from 1 to 4 categorized as the "Low" class and values from 5 to 9 as the "High" class. For Pleasure, 40% of the samples fall into the "High" class, while 60% are categorized as "Low," indicating a tendency towards lower Pleasure levels in this dataset. Arousal is heavily skewed towards the "Low" class, with 97% of samples falling in this range and only 3% in the "High" class, reflecting lower overall arousal in the interactions. Dominance also shows a notable skew, with 77.6% of samples classified as "Low" and only 22.3% as "High," suggesting that participants generally experience lower perceived control in these scenarios.

This distribution indicates that the MITHOS dataset is more unbalanced across the affective dimensions than DEAP, with significant skews, particularly in Arousal and Dominance. The positive correlation between Pleasure and Dominance suggests that as teachers perceive the interaction more positively, they also feel a greater sense of control, which is crucial for managing classroom dynamics.

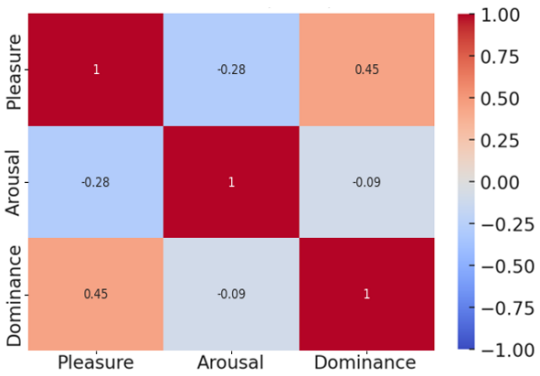


Figure 3.11: Confusion Matrix that shows correlation between the PAD dimensions in the MITHOS dataset

Limitations

The MITHOS dataset includes data from 38 participants. However, only 33 of these were considered valid for analysis due to recording issues with the remaining 5 participants. These issues included instances where the video recordings were missing or instances where incorrect calibration caused misalignment of timestamps across different modalities. Such discrepancies made it impossible to synchronize and analyze the data effectively, leading to their exclusion from the final dataset. One of the primary constraints is its relatively small size, consisting of only 85 minutes of recorded data. This limited amount of data can pose challenges in training machine learning models effectively, as models typically require larger datasets to learn robust and generalizable patterns. To address the issue of limited data size, data augmentation techniques are employed. These techniques help to artificially increase the dataset size and introduce variability, making models more resilient to overfitting. For video data within the MITHOS dataset, we apply augmentation methods such as rotating images by small angles (e.g., ±10 degrees), performing vertical flips, and applying various color transformations. These methods effectively expand the dataset by creating new variations of existing data, helping the model to better generalize to unseen examples.

In the DEAP dataset, we observe significant variance in labels across participants for the same video, as depicted in Fig. 3.5, which highlights the variability and subjective nature of emotional responses among individuals. For the MITHOS dataset, however, a comparable analysis of label variance within specific scenarios is challenging due to the nature of interactions determined by the Wizard-of-Oz (WoZ) setup. The first PAD label per participant is recorded as a baseline measure. Subsequent responses are shaped by how the study master interprets and reacts to each teacher's behavior, with each cue influenced by preceding interactions. Although guidelines are provided to the wizard to promote consistency, there remains an element of subjectivity in modeling student reactions, which may introduce variability in the data. Since these cues vary across scenarios and are not standardized for all participants, a similar analysis of label variance was not conducted. Additionally, even though the simulated nature of these interactions is designed very closely to real life scenarios, they may still not fully capture a few complexities of real classroom dynamics, potentially impacting the ecological validity of the findings.

Chapter 4
Implementation

In this chapter, we will see the machine learning model's architecture we have constructed that can be integrated with our virtual agent. We will first understand the setup and concepts, as they will be used widely throughout our implementation. We will then see the data preprocessing done before inputting them into the model. Our model consists of 3 steps: 1) Feature extraction from the Video Model 2) Feature extraction from the audio model 3) the Fusion model that generates the output. We will look into all these individual parts in the following chapter. Details of why these models were chosen for implementation will be explained in the next chapters with Experiments and Results.

4.1 Setup and Environment Configuration

Docker is a platform that allows the creation and management of lightweight, isolated containers that package code and its dependencies, ensuring consistency across different environments. Using a container simplifies the coding environment, making it easier to reproduce experiments and deploy applications without worrying about system-specific variations. In this setup, we use the *nvcr.io/nvidia/pytorch:22.05-py3* image, which provides a pre-configured environment with PyTorch, CUDA, and other libraries optimized for GPU acceleration, making it ideal for deep learning tasks. We install *opencv-python-headless==4.5.3.56* as it is compatible with the base image and supports face recognition and cropping tasks without requiring a graphical user interface. The *dlib* library is included for efficient face and body detection, leveraging its robust algorithms for identifying and extracting relevant information. Additionally, the transformers library is installed to access a range of pre-trained transformer-based models, which are used throughout our implementation for tasks like feature extraction and emotion recognition. This setup ensures an efficient, consistent, and reproducible environment for our deep learning experiments.

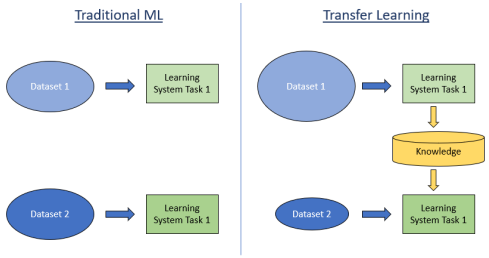


Figure 4.1: Traditional ML models learn each task independently, while Transfer Learning leverages knowledge from previous tasks to improve performance on new, related tasks [44]

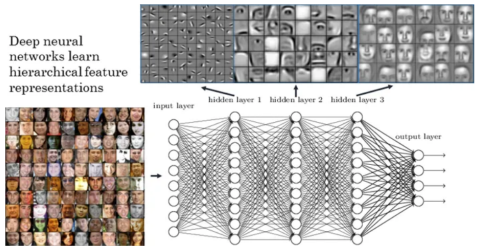


Figure 4.2: Early layers capture simple patterns, while deeper layers recognize complex structures [21], that can be used for Transfer Learning

4.2 Transfer Learning Overview

Traditional machine learning models are typically designed to perform a specific task using a dedicated dataset. For each task, a separate model is trained from scratch, and its performance heavily depends on several factors like the size of the dataset, the quality of the training data, and the amount of training time or resources available. Having looked at the Related Work (Chap. 2) and Datasets(Chap. 3) sections, we know that we will have to implement a deep neural network model for our implementation with relatively small data sizes. These models frequently fail to achieve optimal results when there is insufficient data or limited training resources. They also lack the ability to generalize across other datasets because they are prone to overfitting, meaning they perform well on the training data but struggle to generalize to new, unseen data. To overcome these issues, we widely use the concept of Transfer Learning in multiple sections of our technical implementation of the prediction model. In this section, we introduce and explain the concept behind Transfer Learning, instead of defining its relevance at every point of use.

Transfer Learning allows us to leverage knowledge acquired from one task to improve the performance of a model on a related but different task [44]. The core idea behind Transfer Learning is that models trained on large, diverse datasets (often referred to as source tasks) learn general patterns that are transferable to other tasks (target tasks). This figure 4.1 shows the difference between traditional machine learning, where each task is learned independently from separate datasets, with Transfer Learning, where knowledge gained from learning one task on a dataset is transferred to enhance the learning of a related task on a different dataset. Instead of starting from scratch, a pre-trained model—already familiar with certain features or patterns from a previous task—can be fine-tuned to suit the needs of the new task. For example, if we train a model to recognize basic facial features across a large dataset of faces (Task 1), the model would have learned useful representations, such as edges, eyes, and mouth shapes as shown in Fig. 4.2. The figure 4.2 shows that abstract features do not need to be relearned from scratch for a new task instead can be reused, allowing the model to focus on learning more specific, relevant features for the new dataset, thus accelerating and enhancing the learning process. If we then want to apply this model to a more specific task, such as recognizing emotions or identifying particular facial expressions in a new environment (Task 2), Transfer Learning allows us to reuse the foundational facial features learned in Task 1 to accelerate and improve the learning process for Task 2. This approach is particularly beneficial when

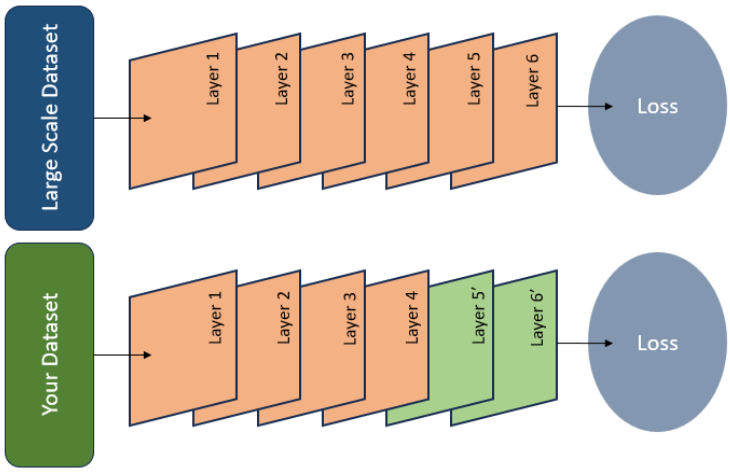


Figure 4.3: Transfer Learning in Action: Initial layers trained on a large-scale dataset are reused, while only the final layers are either fine-tuned or replaced for the new dataset, optimizing model performance with minimal retraining.

there is less data available for Task 2, as the pre-trained model already possesses useful knowledge that can be transferred.

In our case, Transfer Learning involves taking a model pre-trained on a general but similar use case and then fine-tuning it by making it specific for our use-case. Fine-tuning can be done by re-training the latter part of the model's layers that capture the specific features with the new dataset and adjusting the learned weights to adapt to the target task as shown in Figure 4.3. In Figure 4.3, the 5th and 6th layers(in green) show the fine-tuned or replaced layers adjusted to the new dataset. If the pretrained model was originally designed for a task that does not align with our use case, we can replace specific layers to ensure it computes the necessary outputs for our target application eg. replacing a classification head with a regression head, etc. To implement transfer learning, the prior layers of the pre-trained model are often frozen and training is performed only on the new and/or latter layers on top of it to map features to the desired output. [44] [16]

4.3 Model Architecture Overview

In this architecture, we utilize a model fusion technique to process both audio and video data, extracting meaningful features from each modality independently before combining them for prediction. First, audio and video inputs are passed through separate models: an audio model to capture auditory features and a video model to capture visual features. These models are each responsible for generating representations that encode specific characteristics relevant to affect analysis within their respective domains. These audio and video extracted features are then combined with a fusion model. This intermediate model-based fusion approach allows each modality to contribute unique insights, enhancing the model's capacity to identify complex patterns. The fusion model then integrates these modality-specific features to generate predictions for Pleasure, Arousal, and Dominance (PAD) values, allowing for a comprehensive understanding of emotional states based on multimodal data.

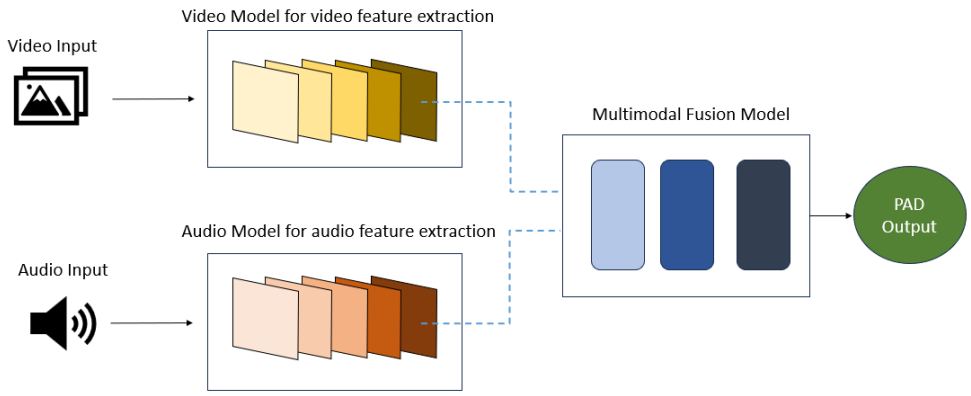


Figure 4.4: Multimodal Fusion Architecture: Individual video (yellow) and audio (orange) models tuned for feature extraction from video and audio inputs, which are then fused in a multimodal fusion model (blue) to predict PAD values.

In the given architecture in Fig 4.4, the video and audio models serve as the feature extractors, providing input to the fusion module. To ensure the fusion module receives high-quality, task-relevant features, we first need to train or fine-tune the video and audio models on our datasets. This ensures that each model learns to extract features that are most relevant information, to predict PAD values in human-virtual agent interactions. Therefore, we need to train each model individually on the dataset to ensure that they are good representations for our fusion module that combines them more effectively and makes accurate predictions. We will follow the below steps to train our model:

1. **Training and Fine-Tuning individual models**: Pretraining the individual audio and video models on the DEAP dataset to learn general characteristics (for video only) and then fine-tuning on the MITHOS dataset (for audio and video).
2. **Feature Extraction**: Once fine-tuned, these models are used to extract high-level features from the video and audio inputs.
3. **Fusion Model**: The extracted features are passed into the fusion module, which learns to combine them, taking into account the complementary information provided by both modalities and outputs the predicted PAD values.

Each of the individual models are elaborated on in the next sections.

4.4 Evaluation

To evaluate our model's performance in predicting Pleasure, Arousal, and Dominance (PAD) values, we employ a range of metrics as explained below that measure prediction accuracy and consistency.

1. **Mean Absolute Error (MAE)**:
   MAE measures the average magnitude of errors between predicted and actual values without considering direction. It is calculated by taking the absolute difference

between predicted and actual values and averaging these differences across all predictions. In this context, MAE helps us understand how far off, on average, our predictions are from the true PAD values. For instance, if the true value of Arousal is 7 and our model predicts 5, the absolute error for that prediction would be 2. A lower MAE indicates better model accuracy, as it reflects smaller deviations from the actual values. MAE is calculated given the below formula where *n* is the total predictions, *yi* is the actual value at *i*th sample, $\hat{y}$ is the predicted value at *i*th sample

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{4.1}$$

2. **Pearson Correlation Coefficient (PCC):**
   PCC assesses the linear relationship between the predicted and actual values, with a range from -1 to 1. A PCC close to 1 indicates a strong positive correlation, meaning that as the actual values increase, the predicted values also increase correspondingly. In the case of PAD prediction, a high PCC implies that the model's predictions closely follow the trends of the actual values. For example, if the true values for Dominance show a rising trend, a high PCC would mean that the model's predictions reflect this trend accurately. PCC is calculated given the below formula where *n* is the total predictions, *yi* is the actual value at *i*th sample, $\hat{y}$ is the predicted value at *i*th sample, $\bar{y}$ is the mean of actual values and $\bar{\hat{y}}$ is the mean of predicted values.

$$\text{PCC} = \frac{\sum_{i=1}^{n}(y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}\sqrt{\sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}})^2}} \tag{4.2}$$

3. **Class-Based Accuracy:**
   It is useful to analyze predictions in broader classes to see if they generally fall into "High" or "Low" categories to help compare our model with other related work. For instance, PAD values could be divided into "High" (values 6–9) and "Low" (values 1–5) classes. Class-based accuracy measures how often the model correctly predicts whether a value belongs to a "High" or "Low" class, rather than predicting the exact value. This metric can give a coarse-grained view of the model's performance and help understand if the model consistently classifies emotions at the correct intensity level.

4. **Test Accuracy within Ranges:**
   This metric checks if the predicted values fall within a specified range around the actual values, here defined as ±2. It is particularly useful for understanding how close the predictions are to the actual values in a practical sense. For example, if the actual value of Pleasure is 6, and the predicted value is 7, it wouldn't necessarily be deemed as incorrect, providing insight into the extent of permissible error. This helps to capture the "tolerance" of predictions and assess the robustness of the model, especially in cases where an exact match isn't critical but staying within a close range is desirable.

5. **Mean Square Error Loss:**
   MSE Loss is used as the primary loss function during model training. MSE measures the average of the squared differences between predicted and actual values, penalizing larger errors more heavily than smaller ones. This makes MSE particularly effective in regression tasks, such as PAD prediction, where reducing larger deviations is essential for improving model accuracy. By minimizing the MSE loss,

the model learns to make predictions that are closer to the true values, improving its overall performance. MSE is calculated given the below formula where n is the number of samples, *yi* is the actual value at *i*th sample, $\hat{y}$ is the predicted value at *i*th sample.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{4.3}$$

4.5 Implementation of model for video feature extraction

We train our video model to ensure that the extracted features align with our task requirements for multimodal fusion. In this section, we outline the technical steps involved in training a video model for feature extraction, which is a key component of our fusion framework. The video data consists of cropped facial data extracted per video with corresponding PAD labels, represented by continuous values from 1 to 9. We use a pre-trained transformer-based model, VideoViT [2], which is adapted and further trained on the DEAP dataset before applying transfer learning on the MITHOS dataset to optimize it for our use case.

4.5.1 Training of DEAP dataset

Data Preprocessing and Loading
The first step is to ensure that the raw video data is preprocessed in a format that is suitable for the model. We use only face data to make predictions as other parts of the video don't contribute to the model's learning. The code uses the Dlib library's get_frontal_face_detector() method to identify faces because emotions are often displayed through facial expressions. The preprocessing pipeline includes detecting faces in each video frame, resizing the detected facial regions, and padding the data to ensure consistent input dimensions. Once a face is detected, the face region is then cropped and resized to the specified target_size 224x224x3 corresponding to the height in pixels, width in pixels and number of channels which is essential to our model. The DEAP dataset has videos with a frame rate of 50 FPS. Processing every frame would be computationally expensive and redundant, as consecutive frames often contain highly similar information, and hence the code skips frames, reducing the frame rate from 50 FPS to 1 FPS, focusing only on significant changes in facial expressions. Every video is padded such that the sequence of frames are of fixed length (60 frames per video), which is crucial for capturing temporal patterns in the data. Now that each video is 60 seconds long, the desired shape for every preprocessed video is 60x224x224x3. We have 18 participants with 40 videos each making the video data of shape 720x224x224x3. This data is then stored in pickle files for easier load. Another pickle is saved that extracts the labels i.e. the PAD values corresponding to 720 videos.

Custom Video Dataset
A custom VideoDataset class is defined, inheriting from torch.utils.data.Dataset. This class encapsulates the video sequences and their corresponding labels. A function "preprocess_frames" was created to reshape video frames into smaller patches, a necessary step when working with pre-trained Video-based transformer models that are designed to operate on patched data. The frames are rearranged from the original

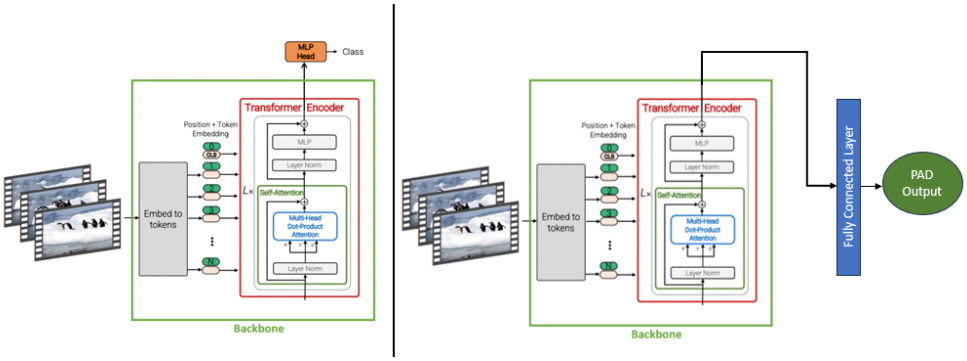


Figure 4.5: Video model architecture used for feature extraction. (Left) Original Video ViT (ViViT) model used for classification tasks. (Right) ViViT model adapted for PAD prediction.

(batch_size, num_frames, channels, height, width) format into patches of size (batch_size, num_patches, patch_size^2 * channels).

By restructuring the video frames in this manner, the model can process each patch in parallel, which helps extract spatio-temporal features. Sampling the frames ensures that the input length remains consistent for each video, maintaining computational efficiency.

Dataset Splitting

To train the model, the dataset is split into training, validation, and test sets. The dataset consisted of 18 participants with 40 videos each making it 720 videos in total. The last 3 participants were considered as the static test set throughout the training and evaluation process (last 120 videos). The remaining 600 videos were shuffled and divided into the train and validation sets in an 80-20 split. We use a KFold cross-validation strategy to ensure that the model's performance is robust and generalizable. This approach ensures that every part of the dataset is used for both training and validation, reducing the risk of overfitting to specific subsets of the data and in turn evaluating completely on unseen data represented by the test set.

Model Architecture

The model used in this implementation is based on the VideoViT (ViViT) architecture, a video-specific transformer originally designed for video classification tasks like playing a guitar, boxing, and much more [2]. It has been pre-trained on large video datasets such as Kinetics-400 or Kinetics-600, which contain thousands of labeled video clips spanning a wide variety of human actions and activities. VideoViT is suited for processing video input as it helps the model focus on learning meaningful representations, and is fine-tuned here for regression of PAD prediction.

The model architecture as shown in Figure 4.5 consists of two main components:

1. *ViViT Backbone*:
   This pretrained transformer processes the video patches to extract spatio-temporal features from the video sequences. The original ViViT model has transformer encoder layers and an MLP head (classification layer) as shown in Figure 4.5 (left).

2. *Fully Connected Layer (FC)*:

The MLP head (classification layer) responsible for predicting tasks in the original ViViT model is replaced with a fully connected layer as shown in Figure 4.5 (right). The output from the ViViT backbone is passed through this new linear FC layer, which maps the extracted features to the final output, i.e., the predicted continuous PAD values. This layer is crucial for converting the high-dimensional feature space into a one-dimensional value suitable for regression.

We freeze the first 4 layers as they have already captured the abstract information about the video data and unfreeze the next 7 layers to learn gerneral features from the DEAP dataset. The ViViT Backbone (shared parameters) remains common for all the PAD dimensions, but the Fully Connected Layer is duplicated for Pleasure, Arousal and Dominance each, for better tuning of the model and performance as shown in Figure 4.6.

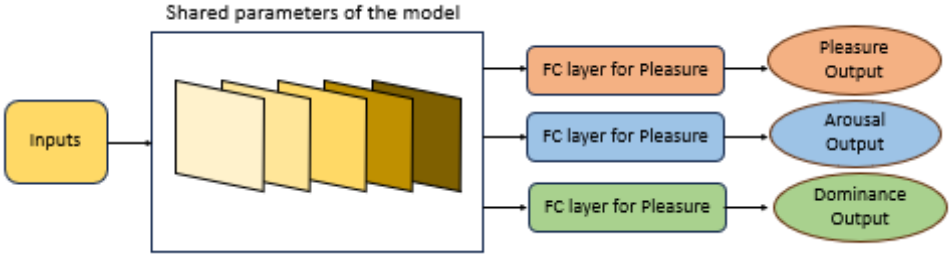


Figure 4.6: Unified model architecture that makes 3 different predictions from these shared parameters

Training Procedure on DEAP data
The training loop is initially defined to optimize the model's weights over 50 epochs. During each epoch, the model's parameters are updated based on the Mean Squared Error (MSE) loss between the predicted and actual PAD values.

For optimization, the Adam optimizer is used. Adam is known for its efficient handling of sparse gradients and adaptability to different types of data. We perform backpropagation i.e. update the weights of the parameters on the fully connected layers (for each dimension) and the last 7 unfrozen encoder layers, while keeping the other layers of the pre-trained backbone frozen to adapt to our inputs. This technique allows the model to focus on fine-tuning the latter layers without altering the learned representations in the base layers, ensuring stable and efficient training.

Validation and Early Stopping
After each epoch, the model's performance is evaluated on the validation set to track its generalization capability. A validation loss lower than the previous best validation loss results in model improvements being retained. An early stopping mechanism is used to prevent overfitting, stopping the training when the validation loss stops improving for five consecutive epochs.

MAE and PCC are also computed to provide additional insight into the model's predictive performance. The class-based accuracy metric is calculated to evaluate the model's ability to differentiate between different PAD values.

Testing and Final Results
Once training is complete, the model is evaluated on the test set, which was held out during training and validation and completely unseen by the training process. The test-

ing procedure includes computing several accuracy metrics like MAE, PCC, Class-Based Accuracy and Accuracy within Ranges for final evaluation. The results for each fold are logged, and the overall performance of the model across all folds is averaged to avoid train-validation split biases, ensuring a comprehensive evaluation of the model's capability.

Saving the model
We now want to use this model pre-trained on the DEAP dataset to fine-tune further on the MITHOS dataset for more task-specific learning and therefore we need to save it. The model was trained using a variety of hyperparameter configurations to evaluate and determine the ideal set that would yield the best performance. Once the perfect hyperparameters (like the number of layers to unfreeze, number of epochs, batch_size, learning rate, etc.) were identified, we trained the model on the entire DEAP dataset (without a train-validation-test split) to fully exploit the available data. The model is trained for 38 epochs using the whole DEAP video data, by keeping the last 7 encoder layers unfrozen with a learning rate of 0.001 and a batch size of 8 sequences suitable for our environment configuration. The model, along with the selected hyperparameters, was then saved with the name *'Pretrained_VideoViT_DEAP.pth'* for future use. This process fine-tuned the model and improved its generalization capabilities making it more context-aware.

4.5.2 Fine-tuning the video model on MITHOS dataset

In the previous section, we see that our video model was tuned on the DEAP dataset to learn general qualities about affect analysis. But we the model to perform better in our use case i.e. predict PAD values human-virtual agent dyadic interactions. To do this, we fine-tune our model pre-trained on the DEAP dataset on the MITHOS dataset. This section focuses on using transfer learning to further refine the pre-trained model to fit the new dataset, MITHOS, which will be used later for feature extraction in a fusion model.

As the data preprocessing, model architecture, and training process are mostly the same as explained the the previous subsection, we will elaborate only on the parts that are different.

Data Preprocessing and Loading

The first step is to ensure that the raw video data is preprocessed in a format that is suitable for the model. The steps that detect faces from the videos and perform other preprocessing remain the same as the ones done for the DEAP dataset. The only different part is that every video in the DEAP dataset is 60 seconds long which remained constant throughout, which doesn't hold for the MITHOS dataset. As every participant's experiment in MITHOS is divided into 6 cues and the videos are also cropped based on these cues. This cropping process is explained in the MITHOS dataset Section 3.2.1, and is performed according to the timestamps present in the Excel file. Unlike the DEAP, the lengths of the MITHOS videos are different so we pad each video with the length of the longest video in order to maintain consistency. We also apply data augmentation techniques such as rotating images by small angles (e.g., ±10 degrees), performing vertical flips, and applying various color transformations to increase the size of the dataset. After preprocessing the final shape of the preprocessed MITHOS data is no_of_frames x 224x224x3 corresponding to the no_of_frames with faces detected extracted from the videos, height in pixels, width in pixels and channels. This is then loaded for model training.

Model Initialization and Training

To train the model, the dataset is split into training, validation, and test sets. In the MITHOS dataset, we have 198 data points. The last 3 participants (10% of valid participants) were considered as the static test set throughout the training and evaluation process (last 18 videos). The remaining 180 videos were shuffled and divided into the train and validation sets in an 80-20 split. We use a KFold cross-validation strategy to ensure that the model's performance is robust and generalizable.

The code then loads a pre-trained model state ('Pretrained_VideoViT_DEAP.pth') saved earlier, which contains the weights optimized for the DEAP dataset. This model serves as the base for fine-tuning on the MITHOS dataset, ensuring that the model benefits from the knowledge acquired during pre-training.

We employ another round of transfer learning for the model to adapt to the MITHOS dataset. Freezing the earlier layers allows the model to retain its learned feature representations from the DEAP dataset, while only adjusting the last encoder layers to adapt to the specifics of the MITHOS dataset. This approach significantly reduces training time and prevents the model from forgetting useful features learned from the previous dataset. The optimizer used is Adam, with a learning rate of 1e-4. The optimizer is tasked with updating the parameters of the unfrozen layers (the last layer and the FC layer), making small adjustments to these layers to fine-tune them for the new dataset. The loss function is Mean Squared Error (MSE) which is aimed to minimize the difference between the predicted and actual values.

As the model is initialized, it is trained on the new dataset with the same process followed in the previous section. An early stopping mechanism is also employed that prevents overfitting and saves computational resources.

Finally, the model is evaluated on the test set, and metrics such as MAE, PCC, Class-Based Accuracy and Accuracy within Ranges are computed to assess the model's final performance.

Saving the model

Initially, we split the MITHOS dataset into training and test sets to identify the optimal hyperparameters for the model. During this phase, the model was trained using a variety of hyperparameter configurations to evaluate and determine the ideal set that would yield the best performance. Once the ideal hyper-parameters were identified, the model was trained on the entire MITHOS dataset (without a train-validation-test split) to fully exploit the available data. The model is trained for 10 epochs using the whole MITHOS video dataset, by keeping the last encoder layer unfrozen with a learning rate of 0.0001 with batch size 4. The model, along with the selected hyper-parameters, was then saved for feature extraction in the fusion model (('Pretrained_VideoViT_DEAP_MITHOS.pth')). This process fine-tuned the model to our use-case of human virtual agent interactions represented in the MITHOS dataset.

Conclusion

The model has followed a comprehensive transfer learning pipeline by utilizing multiple stages of pre-training to achieve optimal performance for video-based feature extraction as shown below:

1. **Pre-trained on a large-scale video transformer model (ViViT)** to learn robust representations for objects, scenes, and temporal dynamics across various domains.

2. **Fine-tuned on the DEAP dataset** to learn specific features relevant to affective state prediction (PAD).
3. **Fine-tuned on the MITHOS dataset** to adapt to human-virtual agent interaction scenarios with high accuracy.

This context-aware model is specialized for the task at hand— PAD values' prediction in dyadic human-virtual agent interactions—while maintaining the flexibility to generalize across other video-based affective tasks to be used in a fusion model. The process leverages transfer learning to efficiently adapt the model, ensuring that the extracted features are suitable for the task while minimizing training time and computational costs. All reasoning that led to the finalization of this approach is explained in the experiments section 5.1.

4.6 Implementation of model for Audio feature extraction

This section provides a detailed explanation of the implementation of an audio model designed for feature extraction in the context of a fusion module. The model aims to predict three emotional dimensions— Pleasure(Valence), Arousal and Dominance values— using audio input data. The pretrained Wav2Vec2-based model designed by Hugging Face utilizes a multi-layer convolutional encoder to transform raw audio into latent representations, which are then passed through a Transformer network to capture long-range dependencies, as shown in Fig. 4.7 [42]. As it has already been trained on large amounts of data to predict PAD values from raw audio signals. It is important to note that we fine-tune this model as we want the best features that represent our use case to input to the fusion model. We use only the MITHOS dataset to fine-tune our model as the DEAP dataset doesn't have audio data in its dataset.

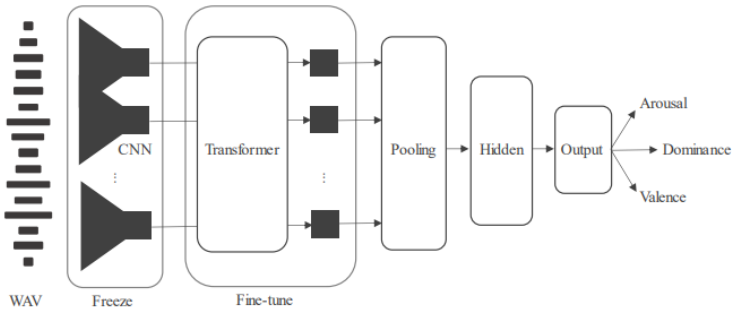


Figure 4.7: Wav2Vec2 model architecture [42] used for processing raw audio signals for prediction

Data Preparation and Preprocessing
The audio repository for MITHOS has audio files in WAV format for every participant in the experiment from start to end. However, as every experiment has 6 instances, we use the Excel file created in the MITHOS dataset that has timestamps for each data instance giving details about the 3 emotional PAD dimensions. The last 3 participants (10% of valid participants) were considered as the static test set throughout the training and evaluation process (last 18 videos). The remaining 180 videos were shuffled and divided

into the train and validation sets in an 80-20 split. We use a KFold cross-validation strategy to ensure that the model's performance is robust and generalizable.

The EmotionDataset class is designed to handle the loading and preprocessing of the audio data. Each audio file is loaded using the Librosa python library and resampled to a consistent sampling rate of 16,000 Hz as it is required for the Wav2Vec2 model as input [42]. Since the audio files may vary in length, a standard approach is applied: if the audio is shorter than the audio of the longest audio signal, it is padded with a constant value i.e. zeros. This ensures uniformity in input length across the dataset, which is important for batch processing in the neural network.

The audio signal is then passed through a Wav2Vec2Processor, which processes the audio signal into a tensor format that is suitable for the Wav2Vec2 model. This step involves converting the raw waveform into discrete tokens that can be used by the model for further feature extraction. This processing step is crucial as the Wav2Vec2 model expects input data in a specific format, and the processor handles this conversion.

Model Architecture
The model architecture used in this implementation is based on Wav2Vec2, a powerful transformer-based model that has been pretrained on large-scale audio data. The architecture allows for the extraction of high-level audio features, which are particularly useful for tasks like speech and emotion recognition.

1. *Wav2Vec2 Backbone:*
   The backbone of the model consists of the pretrained Wav2Vec2 model, which extracts meaningful representations from the input audio. The CNN and Transformer sections of the architecture (Fig. 4.7)capture both the frequency and temporal patterns in the audio signal, which are important for understanding the emotional content of speech. The output of the transformer is then processed using pooling in such a way that we can access the hidden states that could be used for features for the fusion model.

2. *Regression Heads:*
   For PAD predictions, there are three separate RegressionHead layers are used to predict the PAD values, one for each dimension. Each regression head consists of a linear layer followed by ReLU activation and dropout for regularization. The linear layer maps the hidden representations obtained from the Wav2Vec2 backbone to a single output, which represents the predicted value of the respective emotional dimension.

Like the video model, this design ensures that only one unified model can independently learn to predict each emotional dimension from the shared Wav2Vec2 features instead of creating 3 models for P-A-D prediction. The outputs of these heads are the continuous values representing the predicted levels of Pleasure, Arousal and Dominance.

Training Process
The Wav2Vec2 model is well-trained on a large dataset to predict PAD values and therefore we don't need to change the architecture for our use case, instead, just fine-tune it to the MITHOS dataset. The original model predicts values from -1 to 1, but our outputs are from the range 1-9, so we fine-tune the hidden states to suit our model and regression layers to output values from 1-9. The training loop is designed to iteratively optimize the model's weights based on the Mean Squared Error (MSE) loss between the predicted and actual emotional values. The training process proceeds in the following steps:

The model processes the input audio through the Wav2Vec2 backbone, followed by the regression heads, producing three outputs: PAD predictions. Out of the 24 trainable layers, the last layer is unfrozen to learn the nuances in the MITHOS dataset. The MSELoss function is used to calculate the difference between the model's predicted values and the actual values for each emotional dimension. The total loss is the sum of the individual losses for PAD. This ensures that the model is optimized for predicting all three dimensions simultaneously. The loss is then backpropagated through the network, and the model's parameters are updated using the Adam optimizer. The optimizer updates the weights of both the Wav2Vec2 model and the regression heads.

To assess the model's performance during training, MAE, PCC, class-based accuracy and accuracy over a range are computed.

Validation and Early Stopping
After each epoch, the model is evaluated on the validation set to monitor its generalization ability. The validation loss, MAE, and PCC are calculated to track the model's performance over time. Early stopping is implemented to prevent overfitting to the MITHOS dataset. If the validation loss does not improve for a predefined number of epochs (in this case, 5), training is halted early. This mechanism ensures that the model does not continue to train once it has reached its optimal performance, thereby saving computational resources and avoiding overfitting the training data.

Saving the audio model For the audio model, a similar process was followed to identify optimal hyperparameters, with adjustments specific to the characteristics of audio data. The MITHOS audio dataset was initially divided into training and test sets to evaluate various hyperparameter configurations. After selecting the best set of hyperparameters, the model was trained on the complete MITHOS audio dataset (without a train-validation-test split) to maximize data utilization. The model was fine-tuned over 12 epochs, with a learning rate of 0.00005 and batch size 8, keeping the last encoder layer unfrozen. Once trained with these configurations, the audio model and its hyperparameters were saved ('Pretrained_Wav2Vec2_MITHOS.pth') for feature extraction in the fusion model. This process fine-tuned the audio model to the specific requirements of affect analysis in human-virtual agent interactions as represented in the MITHOS dataset.

Test Evaluation
Once training is complete, the model is evaluated on the test set. The test_loader is used to feed unseen audio samples into the model, and the predictions are compared to the actual PAD values. The same metrics— MAE, PCC, and accuracy—are computed to assess the model's performance. This accuracy measure provides insight into the model's precision in predicting continuous emotion values.

Conclusion
The Wav2Vec2-based audio model for emotion regression leverages a powerful transformer-based architecture for extracting robust audio features from raw waveforms. By fine-tuning the model with the regression heads, it learns to predict continuous emotion dimensions from audio data. The implementation includes mechanisms for handling audio input of varying lengths, optimizing model performance with advanced metrics, and ensuring model generalization through early stopping and cross-validation. All reasoning that led to the finalization of this approach is explained in the experiments section 5.2. This approach forms a critical part of the multimodal fusion framework, where audio features will be integrated with other modalities, such as video, to create a

comprehensive emotion analysis system.

4.7 Transformer Fusion Model

This section describes the implementation of a final audio-visual fusion model designed to combine features from both audio and video modalities to predict PAD values. The fusion model is the final stage after feature extraction from the previously trained audio and video models discussed in the previous sections. By leveraging multimodal data, the model aims to enhance the accuracy of the prediction.

The fusion model processes the data in three stages: i) Video feature extraction ii) Audio feature extraction iii) Fusion of the extracted features (from i and ii) through a Transformer-based architecture. Below, we explain each part of the implementation in detail, along with the relevance of the audio and video feature extractors discussed in previous sections.

Preprocessing Input to the Fusion model
In the fusion model, we do not perform explicit preprocessing on the input data, as the inputs consist of feature representations generated by the audio and video models rather than the raw data itself. These extracted features are already preprocessed by their respective models, allowing the fusion model to focus solely on integrating the multimodal information for PAD prediction. Although we don't perform explicit data preprocessing for the Fusion model, we ensure that the preprocessing for the respective audio and video models is carried out in order to generate the feature representations. We also ensure that each data point for both modalities aligns as this structured approach is vital for ensuring that both modalities are handled in tandem during the training process, preserving the temporal and spatial correlations between the two.

Loading the Audio Video models and extracting features
The transformer-based video model (ViViT) and audio model (Wav2Vec2) are initialized and loaded with the pretrained weights saved after tuning them to our dataset for feature extraction. Feature extraction means capturing the output representations from specific layers (in our case, the final layer) that encode important patterns or characteristics learned from the input data. The initial layers of the audio and video models only capture low-level features like objects, faces, sounds, etc. which are more general and don't contribute much to the fusion. Whereas, the last layers contain high-level abstractions that are most relevant to the specific task the model was trained for and provide a good representation of the input data such as visual cues like facial expressions, motion patterns, pitch variations, speech rhythm, and other acoustic cues.

By extracting these domain-specific feature representations, we obtain condensed and meaningful information from the original data, which can then be fed into the fusion model. This enables the fusion model to integrate the unique insights from each modality (audio and video) without needing to process raw data again, which is computationally efficient and effective for multimodal prediction tasks.

Feature projections
The feature representations extracted from the individual models no longer have to predict the PAD values but serve as an input to the Fusion model. The size of the last layers of the ViViT model is of size 768 and that of the Wav2Vec2 is 512. To make them compatible for fusion, the feature vectors have to be of the same dimensionality and

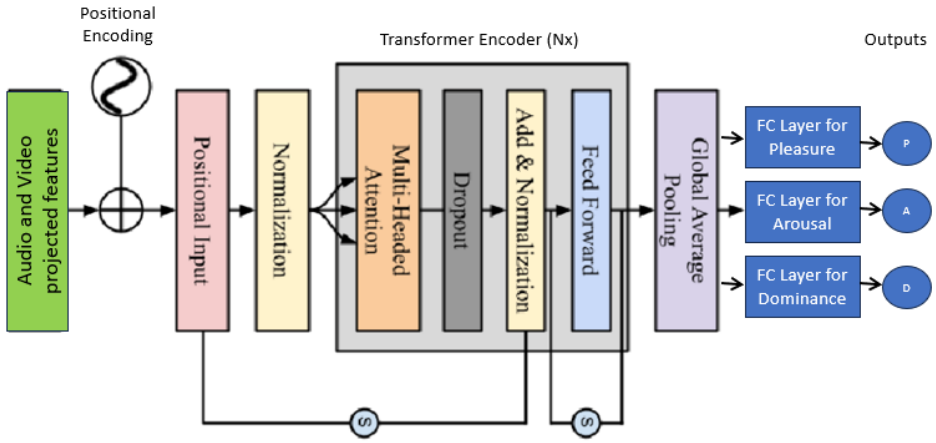


Figure 4.8: Transformer-based Fusion model that takes the projected features extracted by the audio and video models as input and predicts PAD values.

therefore a projection layer is introduced. A projection layer of the audio and video models transforms the feature representations from each model into a common or compatible space. This is typically done by applying a projection layer—often a fully connected layer or a linear transformation—to the output features of each modality of size 32.

Fusion Model Architecture

Once the audio and video features are in the same space, they are fused using a Transformer Encoder as shown in Figure 4.8. This encoder-only setup efficiently captures spatial and temporal dependencies without the additional complexity of a decoder, resulting in a streamlined yet powerful model for PAD prediction.

The Transformer Encoder in the fusion model begins by processing the projected audio and video features, both aligned to a common dimensionality 32 (d_model). Each encoder layer consists of several key components that progressively refine the fused representation. First, the Multi-Head Self-Attention (MHSA) layer splits the input into multiple attention heads (num_heads), each of which computes attention over different subspaces of the feature space, capturing unique dependencies across temporal (audio) and spatial (video) cues; this results in an output dimension of d_model×number_model of_model heads=128, which is then combined and projected back to the size of dmodel. Next, Layer Normalization stabilizes the flow of gradients and keeps the network balanced, ensuring efficient learning across layers. The output is then passed to a Feed-Forward Network (FFN), which consists of two fully connected layers that map the input from dmodel to an expanded space (in our case, 4 times dmodel) before reducing it back to the size of dmodel. With a ReLU activation adding non-linearity to capture complex interactions. Residual connections surround both the MHSA and FFN sub-layers, helping retain the original information while integrating new patterns learned in each layer. This structure enables the Transformer Encoder to build a nuanced, fused representation of the audio and video features, with each layer enhancing the model's ability to capture long-range dependencies essential for accurate PAD prediction.

The transformer's attention mechanism allows the model to weigh the importance of

each modality differently at each time step. For example, in some situations, the video modality (such as facial expressions) might be more informative, while in others, the audio modality (such as tone or pitch) could dominate. The transformer uses multi-head self-attention to compute a contextual representation of the concatenated audio-video features. By learning these cross-modal interactions, the transformer effectively combines information from both the visual and auditory channels, which enhances the model's understanding of the emotional context.

After the Transformer encoder, the fused features are pooled (averaged across dimensions) to obtain a single representation. This representation is then passed through three separate fully connected regression heads—one for each PAD dimension as shown in Figure 4.8. Each regression head maps the pooled features to a continuous score between 1 and 9, representing the predicted PAD values.

Model Training and Testing
The model is trained only on the MITHOS data as DEAP doesn't have audio signals. Audio and Video data is fed in batches through the respective models that extract features for the fusion model which is trained with corresponding emotion labels. The training process for the fusion model involves iterative learning across epochs where only the fusion model's parameters are updated and optimized using a predefined loss function (e.g., MSE). The dataset is split into training and testing data with a 90:10 ratio. The last 18 videos corresponding to the last 3 participants are considered as the test set. The training set is further divided into the train and validation set with an 80-20 percent split that is shuffled across 3 folds for training. During each epoch, the model's performance is evaluated on the validation set to track its generalization ability. To prevent overfitting, early stopping is employed; this technique monitors the validation loss, halting training if the validation loss does not improve for a specified number of consecutive epochs (patience). Early stopping ensures that the model stops training at the point where it performs best on unseen data, rather than continuing to learn specifics of the training set that do not generalize well.

Once training is completed with early stopping, the model is evaluated on the test set to assess its performance on entirely unseen data, providing a final, unbiased measure of accuracy and effectiveness in PAD prediction. This process enables the selection of a robust model that balances accuracy and generalization. The evaluation of the outputs is calculated using MAE, PCC, Class-based accuracy and Range accuracy for each of the dimensions and all of them together.

Conclusion
By combining features from both the audio and video modalities, the fusion model aims to leverage complementary information by using the strengths of each modality and integrating them. For instance, the audio features might capture aspects such as voice pitch, tone, and intensity, while the video features can provide insights into facial expressions and movements. By integrating these two sources of information, the model can make more accurate predictions about emotional states, even when one modality is ambiguous or incomplete making it robust for affect analysis. This approach allows the fusion model to capture complementary aspects of human emotions, resulting in a more comprehensive and accurate emotion analysis.

Chapter 5
Experiments

This chapter consists of all the experiments carried out that led to the architecture decision to refine the final audio-video fusion model for PAD affect prediction. As we know, the model incorporates Wav2Vec2 [42] for audio feature extraction, ViViT[2] for video feature extraction, and a transformer-based fusion model for integrating these multimodal inputs, as shown in Fig. 4.4. The experiments were designed with two primary goals: optimizing each model component individually and evaluating the performance of the fusion model that integrates both audio and video inputs.

This section elaborates on the experiments conducted for each model component — Audio, Video, and Fusion. Each experiment aimed to evaluate specific configurations, hyperparameters, or architectural decisions to determine their impact on predictive performance. The experiments for the audio and video models were conducted independently to optimize feature extraction for each modality, while those for the fusion model explored various strategies for merging these multimodal features to achieve robust prediction.

Please note that the quantitative outcomes of these experiments, such as performance metrics and statistical significance, are explained in the Results chapter (Chap. 6). Here, we focus exclusively on documenting the iterative experimentation process that informed the final model architecture and hyperparameter selections.

5.1 Video Model

5.1.1 Data Preprocessesing Experiments

Preprocessing is a crucial step in the video model pipeline, as it ensures that the input data is consistent with the model to ensure proper training. In the case of video inputs, raw data often comes with challenges such as varying frame rates, resolution inconsistencies, and differences in the length of the videos, across datasets - DEAP and MITHOS. Below are the preprocessing experiments conducted that are aimed at optimizing the learning from video data.

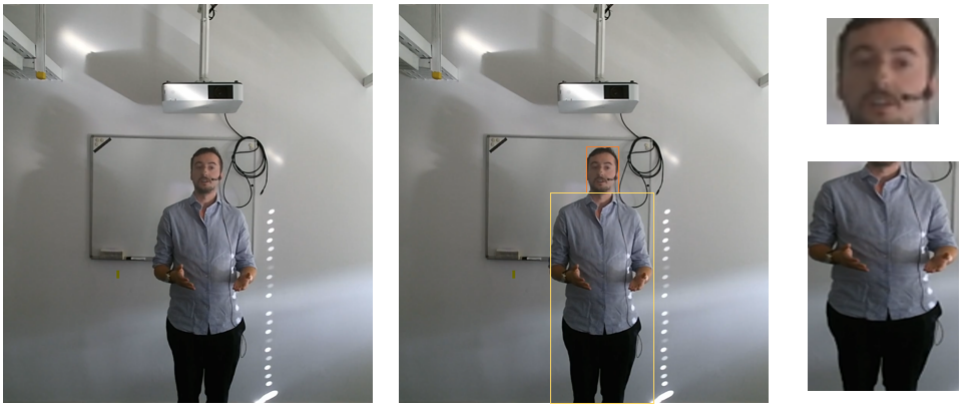


Figure 5.1: Cropping relevant information from a video's frame. (Left) - Whole video frame with redundant information. (Center) - Bounding box around face and body. (Right top) - Cropped face. (Right bottom) - Cropped body.

- **Frame Rate**
  Video data often contains redundant information, especially when recorded at high frame rates such as 50 FPS for DEAP and 25 FPS for MITHOS. Consecutive frames in such videos tend to capture only small or insignificant changes in visual information, which can overwhelm the model with redundant data and increase computational costs unnecessarily.

  Experiment I*: *Choosing the right Frame Rate
  By considering only 1 or 2 frames per second, we can focus on the most significant changes in facial expressions or movements that are relevant to affective state prediction, while reducing the amount of data the model has to process. This not only improves the efficiency of the model by reducing the computational burden but also helps the model learn more meaningful temporal patterns. The chosen frame rate will be consistent for both datasets.

- **Data per Frame**
  From the frames selected for processing, it is essential to focus on extracting relevant information from each frame while ignoring unnecessary background details. For example, as seen in Fig. 5.1 (Left), elements such as the background, door, or equipment in the room are irrelevant to the task of predicting affective states. Instead, our goal is to capture meaningful features like the face, body, facial landmarks, and body movements, as these are the key indicators of emotional expression. By focusing only on these relevant parts of the frame, we reduce noise in the input data and improve the model's ability to learn the important patterns related to human emotions. This also makes the training process more efficient, as the model is not burdened with processing extra information. The different features experimented with per frame are shown below:

  Experiment II*: *Cropped Face from each frame
  Facial expressions are one of the most reliable indicators of emotional states. Features such as eye movements, eyebrow raises, and lip movements provide key insights into the user's emotional responses as shown in Fig. 5.1(Right top)

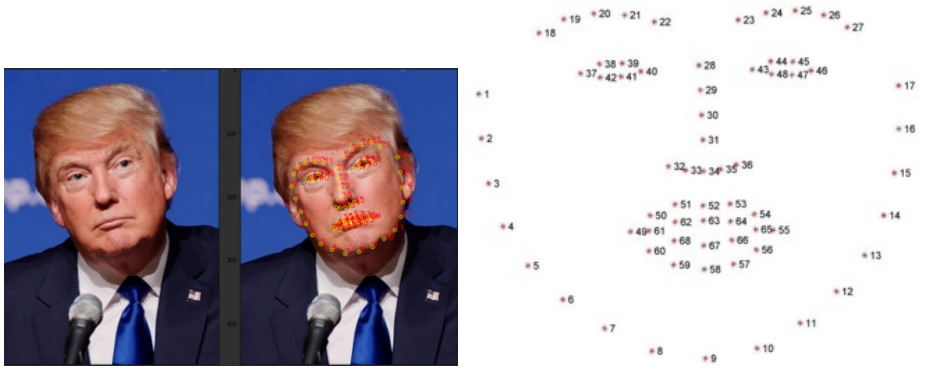


Figure 5.2: Extracting facial landmarks from a cropped face. (Right) Reference image that maps facial landmarks to face. (a) Left - 68 (x,y)-coordinates that map to key facial structures.

The face is cropped from the original image using Python libraries such as Dlib and OpenCV. Dlib's pre-trained facial landmark detector is used to identify the boundaries of the face, and OpenCV processes the detected coordinates to extract the face region by creating a bounding box around the relevant portion of the image and cropping it, as shown in Fig.5.1(Center). By cropping the face, the model is trained on the most significant features related to affective analysis, which enhances its ability to detect subtle changes in emotions without being distracted by unnecessary elements from the background or surroundings.

Experiment III*: *Crop Body from each frame
Body language plays an essential role in conveying emotions. Posture, gestures, and subtle movements like shoulder raises or leaning forward/backward are powerful cues in emotional communication. Cropping the body ensures that these non-verbal signals are prioritized, helping the model detect emotions that may not be fully reflected in the face alone. Fig. 5.1(Right bottom) shows the body is cropped from the original image using Python libraries like OpenCV and Mediapipe. Mediapipe's pose estimation model is used to detect key points on the body, such as the shoulders and hips, to determine the body's bounding box. OpenCV then processes this information by a bounding box to crop the body region from the original image. Cropping the body allows the model to focus on relevant body language, such as posture and gestures, which are important indicators of emotional states.

Experiment IV*: *Extract Facial Landmarks from each frame
Facial landmarks, such as the eyes, nose, mouth, and jawline positions, offer precise geometric information about the face's structure and expression. Tracking these landmarks over time enables the model to identify specific facial movements associated with emotional expressions, such as smiles, frowns, or surprise. This detailed tracking provides the model with a robust way to understand facial dynamics, which enhances its emotional prediction capabilities. The pre-trained facial landmark detector inside the Dlib library estimates the location of 68 (x, y)-coordinates that map to key facial structures as shown in Fig 5.2. While these 68 landmarks comprehensively represent the face, they can overload the model with excessive

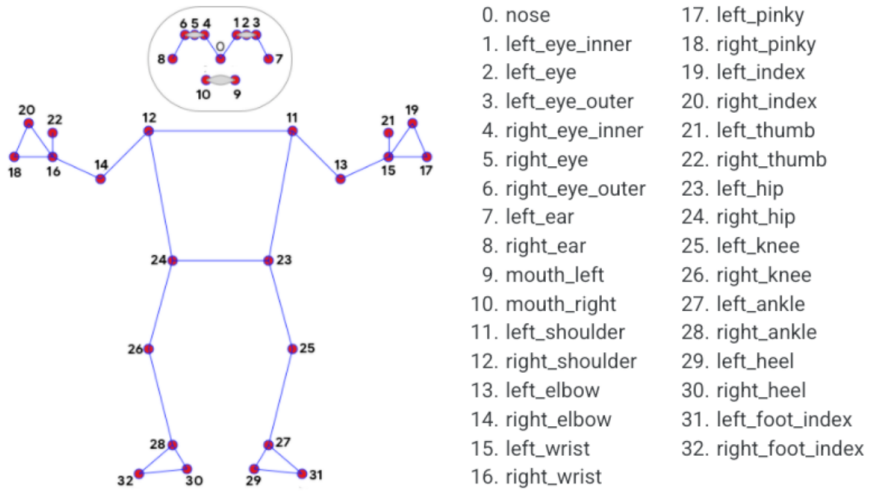


Figure 5.3: Mediapipe's pose estimator that tracks 33 key points on the human body

and often redundant information, especially when subtle facial movements are being analyzed. To reduce this redundancy and improve model efficiency, we select alternate landmarks, which still capture the most relevant facial features without overwhelming the model. This allows the model to focus on the most informative points while reducing computational complexity, ensuring that the network processes only the most critical facial dynamics for affect prediction.

Experiment V*: *Extract Pose Estimates from each frame
Pose estimation involves determining the orientation and movement of the body, including head movements, hand gestures, and overall posture. Pose estimates are crucial for understanding the user's interactions within a scene and detecting more complex affective behaviors such as engagement or frustration. Incorporating pose estimates into the model allows it to better interpret the user's emotional state by analyzing how they move in relation to their environment, adding an additional layer of context to the facial and body cues. Mediapipe's pose estimator is a real-time, machine learning-based framework that detects and tracks 33 key points on the human body, including coordinates for major joints such as the shoulders, elbows, wrists, hips, knees, and ankles, as shown in Fig 5.3. The output is typically a set of (x, y, z) coordinates, where x and y represent the 2D pixel location on the frame, and z represents the relative depth, helping to estimate the 3D pose. This pose estimator can track body movements, gestures, and overall posture, making it useful for tasks like activity recognition, gesture-based interaction, and emotion detection through body language analysis.

- **Time Segments**

Now that the frame rate and feature extraction have been determined, we turn to the length of the video used as input. For the DEAP dataset, the videos are 1 minute long, and each full video is used as a single input, maintaining a consistent length throughout the dataset. In contrast, the MITHOS dataset contains six video snippets per participant, corresponding to different cues logged during the experiment.

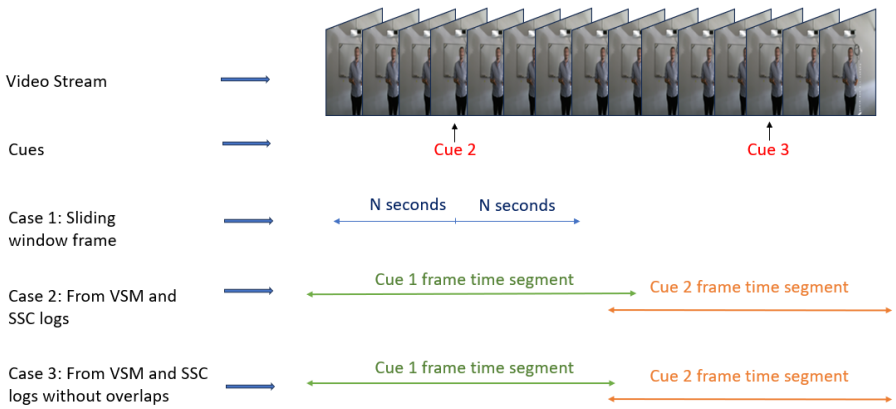


Figure 5.4: Various cases of segmenting for MITHOS data

These video snippets were cropped from the original recordings based on the timestamps provided, with the following condition applied:

Experiment VI*: *Sliding window frame
Experimenting with the moving window frame concept applied in a lot of Related Work. This means that each video snippet spans from 'N' seconds before the cue to 'N' seconds after the cue, capturing a time segment of length '2N' seconds around each experimental event. This sliding window strategy is commonly used in affective studies, as it allows the model to capture the temporal dynamics and subtle changes in emotional states that occur before, during, and after each cue as shown in Fig. 5.4 (Case 1). By employing this approach, we ensure that the model has sufficient context to analyze the participant's affective responses effectively.

Experiment VII*: *Timestamps from VSM and SSC logs The action file logs the various timestamps where actions are recorded by the psychologist during the experiment, based on observed reactions from participants (e.g., phone noise, participant annoyance, etc.). Similarly, the state file logs the different types of interactions initiated by the wizard. For each video snippet, the start timestamp is taken from the action logs, which marks the moment before the cue when the action was triggered as shown in Fig. 5.4 (Case 2). The end timestamp is taken from the state file logs, where the interaction is concluded by the wizard. This approach ensures that the video snippets capture the entire sequence of events, from the trigger action to the conclusion of the interaction, providing a complete context for analysis.

Experiment VIII*: *Timestamps from VSM and SSC logs without overlap
This condition is an extension to the previous condition because there could be instances where the end time of one cue overlaps with the start time of the next cue, resulting in two overlapping snippets with different labels. To avoid this overlap, we manually adjust the end timestamp of the first cue to one second before the start of the second cue, provided that the first cue remains within the range of the first snippet. If the first cue extends beyond the non-overlapping range, we manually shift the start time of the second cue to one second after the end of the first cue. This preprocessing step ensures that there is no overlap between video snippets,

maintaining the integrity of the labeled data for each cue as shown in Fig. 5.4 (Case 3)

5.1.2 Model Selection

Now that we have analyzed the different kinds of data preprocessing, we see which machine learning model captures data the best. Different models are used for experimentation to choose the best that fits our use case.

Experiment IX*: *CNN Model
We use the CNN model as the baseline for this set of experiments. CNNs are well-suited for tasks involving spatial features like facial landmarks and body pose because they are designed to detect and extract local patterns such as edges, shapes, and textures. CNNs have also been extensively used in affective computing and emotion recognition tasks, where they have proven to achieve solid results even with relatively smaller datasets. Additionally, CNNs are computationally less expensive compared to Transformer models, making them a practical choice for a baseline.
Model Architecture
The CNN model used here is a 3D convolutional neural network (CNN) designed to process video input and predict continuous value PAD values. The architecture starts with two 3D convolutional layers (conv1 and conv2), which are responsible for extracting spatial and temporal features from the video. The first convolutional layer (conv1) applies 16 filters, each with a kernel size of 3x3x3, over the input data, capturing localized patterns in both the spatial dimensions (height and width of each frame) and the temporal dimension (across consecutive frames). The second convolutional layer (conv2) further refines these patterns, increasing the number of filters to 32, enabling the model to learn more complex features as it goes deeper. After each convolutional layer, a ReLU activation function is applied to introduce non-linearity, helping the model capture more complex relationships. A 3D max-pooling layer follows each convolution, which reduces the spatial and temporal dimensions, downsampling the feature maps while retaining important information, making the model more computationally efficient.

Once the 3D convolutions and pooling layers have extracted meaningful features, the output is flattened into a 1D vector and passed through a fully connected layer (fc1), which maps the high-dimensional features to a 64-dimensional space, learning a more abstract representation of the data. Finally, another fully connected layer (fc2) reduces the output to a single regression value, corresponding to the PAD dimension in the PAD (Pleasure, Arousal, Dominance) framework. This design ensures that the model can capture both the spatial relationships within each frame (facial expressions, body movements) and the temporal dependencies between frames (how emotions evolve over time), making it a robust architecture for video-based affective prediction.

Experiment X*: *Pretrained VideoResNet
Given the promising performance of ResNet (a CNN-based model), in various applications, we conducted experiments using a pre-trained VideoResNet model. The key advantage of using a pre-trained VideoResNet model is that it has already been trained on large-scale video datasets that allows the model to learn a wide range of relevant visual features, such as spatial structures and temporal dynamics, and other important video-related aspects such as facial structures, motion, etc. This highlights the benefit of using pre-trained models, as they provide a significant advantage in terms of feature

extraction without requiring the design of a model from scratch.

Model Architecture

The VideoResNet model, specifically the r3d_18 version, is a 3D convolutional neural network (CNN) designed to handle video input. The model extends the classic 2D ResNet architecture by incorporating 3D convolutions, which allow it to capture both spatial (within each frame) and temporal (across consecutive frames) features from video data. The 3D convolutions process the height, width, and time dimensions simultaneously, making it ideal for tasks where motion or changes over time are important. The ResNet architecture uses residual blocks, which help prevent the vanishing gradient problem, allowing deeper networks to train effectively. These residual blocks pass information across layers via shortcut connections, ensuring that the model can learn hierarchical features without losing important information as the network deepens.

The base architecture consists of four main residual blocks (layer1, layer2, layer3, layer4), each of which consists of 2 convolutional layers and shortcut connections, each of which extracts increasingly complex features from the video frames. Each convolutional layer in these blocks is followed by batch normalization that normalizes the output of a layer by scaling and shifting the data, and ReLU activation that introduces non-linearity to the model, allowing it to learn complex patterns, with skip connections that help maintain the flow of information across layers. The final fully connected layer of the pre-trained model outputs class predictions for a classification task typically used for action recognition (e.g., running, swimming, dancing). Since we are focusing on predicting a PAD value, we replace the final fully connected layer with a linear layer that outputs a single continuous value between 1-9. Specifically, the final linear layer (fc) is adjusted to have an input size of 512 (the output from the last residual block) and an output size of 1, which predicts the continuous PAD value.

Based on the strong performance of the VideoResNet model, we continue to experiment with other pre-trained video models in subsequent experiments.

Experiment XI*: *Custom Transformer Model

Transformer-based models have demonstrated superior performance over CNNs in handling tasks that require modeling long-range dependencies and complex temporal relationships, making them well-suited for affect prediction in video data. Unlike CNNs, which primarily capture local spatial patterns, Transformers employ self-attention mechanisms to model interactions across entire video sequences, allowing the model to focus on relevant areas both within and across frames. The Transformer Encoder is ideal for our task as it captures critical spatiotemporal relationships without the additional computational load of a decoder.

Model Architecture

Our custom Transformer Encoder model processes video sequences by dividing frames into patches, embedding these patches, and adding position encodings to maintain temporal order. Input frames are divided into patches, embedded, and assigned position encodings to maintain temporal coherence. After encoding the inputs, they are passed through a number of stacked Transformer Encoder Layers. Each encoder layer consists of several key components like the Self-Attention Layer, Position Encoding and Feed-Forward Network (FFN) [40]. The Self-Attention layer enables the model to weigh different parts of the input sequence, which allows them to focus on the most relevant parts, identifying important spatial and temporal relationships essential for PAD prediction. Positional encoding is added to each input token, position encoding preserves the sequence order of frames, which is critical for understanding the evolution of emotions over time. After self-attention, the FFN applies non-linear transformations, enhancing

the model's ability to capture complex patterns within the data. After encoding, the output is pooled and passed through a regression layer to predict a continuous PAD value between 1-9. This design allows the model to capture both spatial and temporal dependencies, making it more effective than CNNs for affect prediction in video data.

Experiment XII*: *Pretrained ViViT Model
Recognizing the advantages of pretrained transformer architectures, we conducted experiments using the Video Vision Transformer (ViViT) [2], a model that has shown promising results in video-based tasks. Unlike a standard Transformer model trained from scratch, the ViViT model benefits from pretraining on large video datasets like Kinetics-400, allowing it to capture complex visual and temporal features directly relevant to input across the entire video, regardless of how far apart those features are, making them well-suited for complex patterns like emotions that may manifest gradually or across different parts of the video across both space and time. This pretrained foundation significantly reduces the computational load and the need for extensive training on our dataset, as the model has already learned general spatiotemporal patterns.

Model Architecture
ViViT was designed by Google Research in 2021 for video classification tasks. ViViT's architecture is tailored for video-based tasks, with a focus on capturing dynamic spatial and temporal features. As it is an extension of the Transformer model, the architectural layers are almost the same. Video frames are divided into patches, which are embedded and assigned position encodings, allowing the model to process sequences of patches across frames. Multiple layers of multi-head self-attention and feed-forward networks process spatiotemporal information, focusing on relationships both within frames and across frames. Transfer learning is applied by leveraging the pretrained weights, ViViT provides a head start in identifying motion, object structures, and temporal dependencies. The input to this new layer is the output from the transformer's final encoder layer, which processes the entire sequence of video patches and aggregates the information necessary for the prediction. The classification layer (originally used for predicting video categories) is replaced with a fully connected layer to output a continuous PAD value for our regression task. The final encoder output is pooled and processed by this layer, allowing for precise PAD prediction.

By using ViViT's pretrained capabilities and adapting its output layer for PAD regression, we efficiently capture the local and global dependencies essential for affective state prediction. This model is particularly advantageous as it combines the benefits of transformer-based architectures with pre-learned video features, making it more robust and computationally efficient for video affect prediction compared to custom models trained from scratch.

Experiment XIII*: *Pretrained SwinTransformer Model
In this set of experiments, we use the Pretrained Video SwinTransformer model. Unlike traditional Vision Transformers (ViTs), the SwinTransformer employs a shifted window mechanism, which processes input patches in overlapping windows across both spatial and temporal dimensions. This approach enables the model to capture local and global context more effectively by gradually increasing the receptive field as the network deepens, without needing to compute attention over the entire image or sequence at once. This hierarchical design allows the SwinTransformer to be more computationally efficient while still preserving important structural and temporal relationships within the video data. Its shifted window attention mechanism allows the model to focus on both local

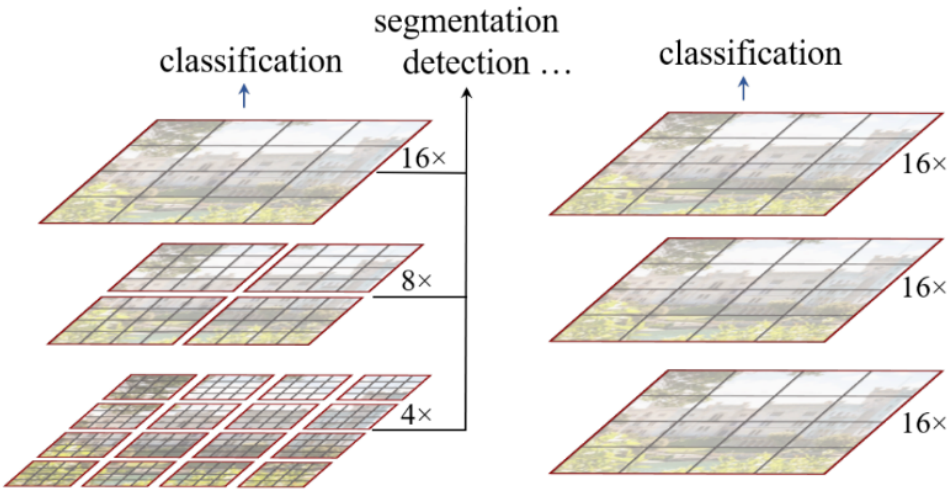


Figure 5.5: Comparison of Swin Transformer and ViT Architectures: (Left)) Swin Transformer processes images in progressively smaller patches as the layers progress (Right) ViT processes fixed-size (whole image) patches for all the layers.

facial details (like eye movements or lip movements) and broader temporal patterns (like changes in expression over time) and its ability to capture multi-scale features through hierarchical attention could offer more nuanced insights into emotional states.

Model Architecture
The SwinTransformer is built on a modified architecture compared to the standard Vision Transformer (ViT). While both models process input images by splitting them into patches, the SwinTransformer introduces a shifted window attention mechanism that improves how spatial and temporal information is captured [24]. The SwinTransformer uses local windows to compute attention, meaning it applies self-attention within smaller, non-overlapping windows. These windows "shift" or move at each layer, allowing the model to progressively integrate information from adjacent regions. This hierarchical approach helps SwinTransformer capture both local and global features more efficiently as the receptive field expands through layers. The architecture consists of multiple SwinTransformer blocks, each containing two main parts:

1. *Window-based multi-head self-attention (W-MSA)*: This restricts attention computation to local patches within a window, making it computationally efficient for high-resolution inputs.

2. *Shifted window-based multi-head self-attention (SW-MSA)*: In this phase, the attention windows shift slightly, allowing connections between patches in neighboring windows to gradually learn global context while maintaining local feature refinement.

The hierarchical design of the SwinTransformer progressively increases the resolution of feature maps as the network deepens. This makes it effective in multi-scale feature learning. We replace the final classification layer with a custom fully connected (FC) layer like done in the ViViT model. The comparison between the SwinTransformer and ViT model processing patches across layers is shown in Figure. 5.5. By leveraging the SwinTransformer's ability to handle both local and global dependencies efficiently, com-

bined with the fine-tuned regression layer, this architecture is well-suited for emotion prediction from video data.

5.1.3 Conclusion

In summary, a range of experiments were conducted to optimize the video model, including tests on data preprocessing and model architectures. These experiments focused on refining the video feature extraction process, ensuring that relevant facial, body, and pose information was effectively captured and utilized for PAD prediction. By exploring various preprocessing configurations and architectural options, the final video model was selected based on its ability to balance computational efficiency and predictive accuracy.

The final architecture (elaborated in Implementation Section 4.5) demonstrated strong performance in extracting affective signals, and its features will be leveraged as input to the fusion model, as detailed in the coming sections. This integration allows the fusion model to benefit from the optimized video features, enhancing its capacity to predict PAD values in conjunction with audio data for a more comprehensive multimodal analysis.

5.2 Audio Model

5.2.1 Data Preprocessing Experiments

The DEAP dataset does not include the audio modality, so audio signals are obtained solely from the MITHOS dataset. For preprocessing the audio data, we use Python's Librosa library to extract the raw audio signals from the experiments. Librosa processes the audio by resampling it to a consistent frequency, typically 16 kHz, ensuring uniformity across all samples. We run the same experiments Exp ***VI, VII, VIII*** and Fig. 5.4 for the lengths of the audio signal based on the cue and therefore won't be elaborated again. However, the audio signals vary in length, which can be challenging for the model to process. To address this, we pad all audio signals to match the length of the longest file using a constant value, ensuring that all signals are of consistent length and can be fed into the model seamlessly.

5.2.2 Model Selection

Experiment XIV: ***CNN Model***
The first experiment involved using a Convolutional Neural Network (CNN) as a baseline model for predicting PAD values from audio signals. The CNN was designed to extract features from raw audio data, capturing key aspects such as speech patterns, tone, and temporal dynamics.

Model Architecture
The CNN model's architecture consists of two 1D convolutional layers (conv1 and conv2), which are crucial for extracting features from sequential audio data by applying filters to detect local patterns like pitch or tone changes. The first layer (conv1) has 16 output channels with a kernel size of 3, capturing small, localized audio features, while the second layer (conv2) increases the depth to 32 channels, allowing the model

to learn more complex patterns. After each convolution, a ReLU activation function is applied to introduce non-linearity, enabling the model to capture more complex relationships in the data. The output is then flattened into a 2D tensor to be processed by fully connected layers (fc1 and fc2). The first fully connected layer (fc1) projects the extracted features into a 64-dimensional space, allowing the model to learn high-level representations, while the second fully connected layer (fc2) outputs a single value for regression, predicting the continuous Pleasure value. This design leverages convolutional layers for feature extraction and fully connected layers for decision-making, providing a structured approach to predicting affective values from audio data.

The simple CNN is a deep learning model and needs a lot of data to train and avoid overfitting so we need a model that is pre-trained on something similar to learn patterns in data without too much volume.

Experiment XV*: *Wav2Vec2 Model
To improve upon the baseline, the Wav2Vec2 model was introduced. Wav2Vec2 was chosen for its superior ability to extract high-level features from raw audio inputs, particularly those related to speech and tone, which are critical for affective analysis. Pretrained on large-scale audio datasets, Wav2Vec2 has been shown to effectively learn representations from speech, making it a strong candidate for predicting PAD values.
Model Architecture
The Wav2Vec2 model, developed by Hugging Face, is a transformer-based architecture designed specifically for extracting meaningful features from raw audio inputs, particularly speech, and it excels in tasks such as speech recognition and emotion analysis [42]. The model begins with a multi-layer convolutional encoder that processes raw audio signals into latent representations. These convolutional layers act as feature extractors that convert the input audio waveform into discrete units, which capture short-term patterns such as speech phonemes or tonal shifts. After this encoding, the latent representations are fed into a transformer network, which is the core of Wav2Vec2. The transformer consists of multiple self-attention layers that capture long-range dependencies and relationships within the audio signal. This is particularly important for affective analysis, as emotions often unfold over longer periods and require contextual understanding. The self-attention mechanism enables the model to weigh different parts of the audio sequence based on their relevance to the task, allowing it to focus on the most meaningful aspects of speech and tone for predicting emotional states. Finally, a regression head is added to the model to map the high-dimensional features extracted by the transformer to the continuous values corresponding to Pleasure, Arousal, and Dominance (PAD). This layer ensures that the final output aligns with the specific affective dimensions required for the task. 4.7

The model's design decisions reflect its ability to handle complex audio data efficiently. The convolutional layers are essential for transforming the raw waveform into a structured input, while the transformer layers are critical for capturing both local and global dependencies in the audio. The use of a pre-trained Wav2Vec2 model fine-tuned on the dataset, ensures that the model can leverage large-scale learned representations while adapting to the task-specific nuances of affect prediction. The final regression head allows for accurate mapping of the learned features to continuous affective values, making Wav2Vec2 a powerful tool for extracting and interpreting emotional cues from audio.

5.2.3 Conclusion

Given the strong fit of Wav2Vec2 for our needs, there was a limited necessity to experiment with multiple alternative architectures or configurations for the audio model. The model's pre-trained structure and fine-tuning potential aligned well with the demands of our task, allowing it to capture relevant auditory patterns with minimal adjustments to fit the MITHOS dataset. The optimized configuration provided robust audio features, making it an essential component of the fusion model for multimodal affect prediction.

The final architecture (elaborated in Implementation Section 4.6) demonstrated strong performance in extracting affective signals, and its features will be leveraged as input to the fusion model, as detailed in the coming sections. This integration allows the fusion model to benefit from the optimized audio features, enhancing its capacity to predict PAD values in conjunction with video data for a more comprehensive multimodal analysis.

5.3 Fusion Model

5.3.1 Data Preprocessesing Experiments

In the case of our fusion model, there is no additional, fusion-specific data preprocessing required beyond what was already performed for the individual audio and video models. Since the fusion model directly takes the extracted features from the audio and video models as input, the preprocessing steps for raw audio and video data—such as frame selection, feature extraction, and temporal alignment—are handled at the individual model level. This simplifies the fusion model's preprocessing requirements, as we rely entirely on the preprocessed features generated by the Wav2Vec2 and ViViT models.

The only requirement for the fusion model is ensuring that the dataset contains an equal number of audio and video feature points for each sample, enabling accurate mapping across modalities. This consistency ensures that each audio feature vector corresponds correctly to its paired video feature vector, maintaining data integrity and alignment across both modalities throughout the fusion process.

5.3.2 Model Selection and Experiments

Experiment XVI*: *Simple (Early) Fusion Model
In this experiment, we explore a basic early fusion approach to predict PAD (Pleasure, Arousal, Dominance) values by concatenating raw video and audio inputs at the input level. This approach allows the model to learn joint representations from both modalities simultaneously, potentially capturing cross-modal relationships that are crucial for understanding affect. By feeding the concatenated inputs into a CNN model, we aim to simplify the architecture while leveraging the combined information from visual and auditory cues to improve prediction accuracies.

Model Architecture
The model architecture for this early fusion approach begins with concatenated video and audio inputs, forming a single multimodal input tensor that captures both visual and auditory information. This input is processed by two 3D convolutional layers: the first (Conv1) applies 16 filters with a 3x3x3 kernel, capturing foundational spatial and temporal features, followed by a ReLU activation for non-linearity. The second

convolutional layer (Conv2) uses 32 filters to detect more complex patterns. After each convolutional layer, a 3D max-pooling layer is applied to downsample the feature maps, reducing dimensionality while retaining essential information. The output from the convolutional layers is then flattened into a 1D vector, which is fed into a fully connected layer with 64 units, learning an abstract representation of the fused input. A final fully connected layer reduces the output to a single regression value, representing the PAD prediction, leveraging the combined multimodal features for affective analysis.

While straightforward, the SimpleFusionModel lacks the capacity to model complex interactions between audio and video features. Its simple concatenation and shallow architecture may fail to capture the nuanced dependencies within and between modalities, resulting in lower predictive performance. These are the downsides of using Early Fusion as mentioned in the Related Work. Additionally, it does not leverage temporal dependencies in the data, which are critical for PAD prediction in dynamic, time-sequenced interactions.

Experiment XVII*: *Cross modal attention Model
In this experiment, we implement a late fusion approach combined with a cross-modal attention mechanism to predict PAD (Pleasure, Arousal, Dominance) values. Unlike early fusion, where audio and video inputs are combined at the input level, late fusion first processes each modality independently, extracting distinct features from both audio and video before merging them. By applying a cross-modal attention mechanism, this approach allows the model to dynamically focus on the most relevant features from each modality in relation to the other, enhancing the model's ability to capture interdependencies between audio and visual cues that contribute to affective states. This enables the model to more effectively leverage information from each modality, especially when certain cues in one modality complement or reinforce cues in the other, leading to more accurate and context-aware PAD predictions.
Model Architecture
The model architecture begins with extracting features for audio from the Wav2Vec2 model and video from the ViViT model. The extracted features from each modality are then passed into a cross-modal attention module as shown in Figure 5.6 (left). In this module, the audio features act as queries while the video features serve as keys and values, allowing the model to selectively focus on video information based on the audio context and vice versa. This cross-modal attention mechanism dynamically weights each feature according to its relevance to the other modality, allowing the model to highlight complementary information and strengthen relevant signals across modalities.

Following the cross-modal attention module, the attended audio and video features are concatenated and passed through a series of fully connected layers. The first fully connected layer maps the concatenated features to a 64-dimensional space, learning an integrated representation of the attended multimodal features. A final fully connected layer then outputs a single regression value corresponding to the PAD dimension. This late fusion approach with cross-modal attention enhances the model's ability to interpret complex, multimodal cues, producing contextually sensitive predictions for affect analysis.

Although this approach gives insights into the cross-modal interaction, it is relatively simplistic and limited in its ability to capture temporal dependencies or model more complex relationships between the modalities. This approach might not fully exploit complementary information from each modality, leading to suboptimal performance, especially in situations where temporal sequencing or dynamic features are essential for

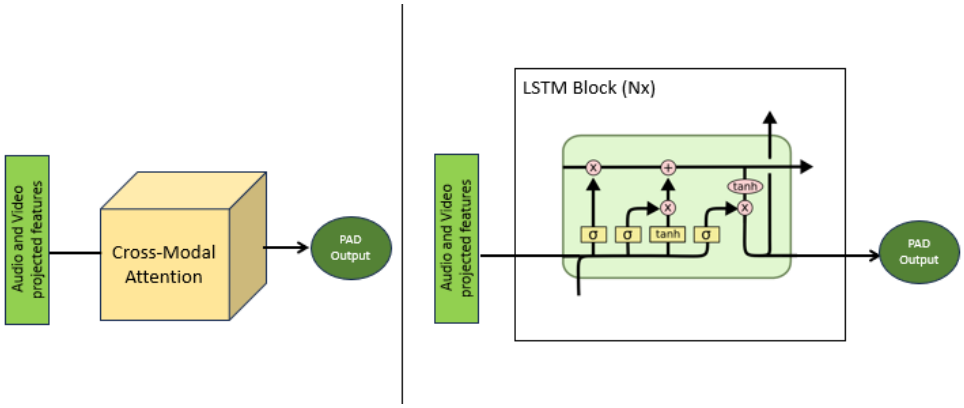


Figure 5.6: Late fusion model experiment's architectures that take audio and video features extracted from the respective models as input. (Left) Cross-Modal attention architecture. (Right) Recurrent Neural Network-based LSTM architecture.

accurate PAD prediction.

Experiment XVIII*: *RNNFusionModel (FNN Fusion Model)

This model was designed to explore the potential benefits of using temporal modeling to capture the sequential dynamics in PAD prediction. The LSTM architecture is well-suited for time-series data, and we aimed to leverage its ability to model temporal relationships in the fused feature space, which could enhance the model's responsiveness to changing affective states over time.

Model Architecture The RNNFusionModel also referred to as the FNNFusionModel, incorporates an LSTM block to capture temporal dependencies between the fused audio and video features as shown in Figure 5.6 (right). The model concatenates the audio and video feature vectors and projects the combined features to a lower-dimensional space, which serves as input to the LSTM layer. The final hidden state from the LSTM is then passed through a fully connected layer to produce the PAD prediction.

The model architecture begins with feature extraction from the respective video and audio models, capturing essential cues from each modality. These extracted features are then passed through an input projection layer, to reduce their combined dimensionality for more efficient LSTM processing. Next, the LSTM layer is employed to capture temporal dependencies within the fused features, allowing the model to learn patterns over time. Finally, regression heads produce the PAD outputs, yielding the predicted PAD values based on temporal and multimodal information.

Although the LSTM layer provides temporal modeling capabilities, the RNNFusion-Model may still fall short in capturing the complex dependencies across both modalities. RNNs can struggle with vanishing gradients and require extensive computational resources when trained on large datasets with long temporal dependencies. Additionally, the model's reliance on sequential processing limits its ability to parallelize effectively, which can slow down training and inference in real-time applications.

Experiment XIX*: *Transformer based Fusion model

After experimenting with various fusion models, we developed a custom transformer-based fusion model to address the limitations identified in previous architectures. This

model is designed to take feature representations from audio and video inputs, extracted by the Wav2Vec2 and ViViT models, respectively, and fuse them into a unified representation for predicting PAD (Pleasure, Arousal, Dominance) values. The transformer's strengths in modeling both spatial and temporal dependencies make it a particularly effective choice for capturing the intricate, dynamic relationships between multimodal affective cues, ultimately yielding superior performance over simpler fusion methods.

One key feature of our custom transformer-based fusion model is its use of a transformer encoder, without a decoder, to process the fused audio-video features. This choice is motivated by the nature of the PAD prediction task, which does not require sequential generation of outputs (as in language generation) but rather a focused, unified representation that combines and interprets multimodal inputs. This encoder-only setup efficiently captures spatial and temporal dependencies without the additional complexity of a decoder, resulting in a streamlined yet powerful model for PAD prediction.

Model Architecture

The custom transformer-based fusion model, illustrated in Figure 4.8, comprises several key components:

1. Input Embedding and Projection: The extracted features from the Wav2Vec2 (audio) and ViViT (video) models serve as inputs to the transformer. To facilitate fusion, the audio and video features are concatenated along the feature dimension and projected to a unified embedding space. This projection ensures compatibility with the transformer encoder's input requirements and aligns the features spatially and temporally for fusion processing.

2. Positional Encoding: Given the sequential nature of video frames and audio segments, positional encodings are added to the input embeddings. These encodings introduce temporal order into the feature representations, allowing the model to distinguish between different positions within the sequence. This temporal ordering is crucial for PAD prediction, where changes over time (such as tone, pace, or body language) can influence emotional interpretation.

3. Transformer Encoder: The core of the model is a transformer encoder consisting of multiple self-attention layers and feed-forward networks. Each self-attention layer enables the model to capture dependencies within and across modalities by attending to all positions in the sequence, both spatially (e.g., individual frames or audio segments) and temporally (e.g., across the entire sequence). This self-attention mechanism enables the model to capture contextual relationships that are essential for interpreting complex affective states.

    - Self-Attention Layers: Each layer performs multi-head self-attention, allowing the model to learn distinct patterns and relationships across various aspects of the data simultaneously.
    - Feed-Forward Layers: These layers introduce non-linearity, helping the model to learn complex transformations and contribute to the expressive power of the transformer for PAD prediction.

4. Pooling and Regression Heads: After passing through the transformer layers, the output representations are pooled to create a single, unified feature vector that captures the relevant information from both audio and video inputs. This pooled feature is then passed through 3 different regression heads (fully connected layer) to predict the PAD values. The pooling step ensures that the model distills the

information into a compact form that is well-suited for predicting continuous values for Pleasure, Arousal, and Dominance.

In summary, the Custom Transformer fusion model effectively integrates audio and video modalities, leveraging self-attention mechanisms to capture complex, cross-modal dependencies and temporal patterns essential for accurate PAD prediction. By dynamically focusing on relevant features from each modality, this model enhances the interpretability and robustness of affective predictions, making it a powerful approach for multimodal affect analysis in human-virtual agent interactions.

5.3.3 Conclusion

This model chosen, following extensive experimentation with various fusion approaches, represents the optimized architecture for predicting the actual PAD values. It effectively combines multimodal inputs in the form of feature representations, dynamically capturing nuanced emotional cues across audio and video, to deliver accurate and context-aware affective predictions.

5.4 Other General Experiments

5.4.1 Model Hyperparameters

The performance of machine learning models, particularly complex models like transformers, is highly sensitive to hyperparameter tuning. For each model component — audio, video, and fusion — several key hyperparameters were tested to determine the optimal configuration that would balance training efficiency and predictive accuracy. Below are the hyperparameter experiments conducted to refine the architecture of the final model.

Experiment XX*: *Number of Trainable Layers
When transfer learning is applied from pretrained models, choosing the number of layers to be fine-tuned can significantly impact model performance. Fine-tuning all layers can capture deeper features but also risks overfitting, especially with smaller datasets. Conversely, training fewer layers can reduce overfitting but may limit the model's ability to capture complex patterns. Freezing layers preserves the general features learned from the pretrained models, focusing only on fine-tuning the layers directly relevant to PAD prediction. However, experiments were also conducted by progressively unfreezing layers, starting from the final layer (detailed features) and moving backward (general features) to understand the impact of additional trainable layers. Results showed the optimal balance between accuracy and training time when fine-tuning a subset of the final layers, which avoided overfitting while improving model adaptability to the PAD prediction task.

Experiment XXI*: *Epochs
Epochs determine the number of times the model iterates over the complete dataset, directly affecting training time and model convergence. To identify the optimal number of epochs, the model was initially trained with a high epoch count, monitoring for overfitting and convergence plateaus on validation loss. Early stopping techniques were

applied to dynamically halt training if the validation performance ceased improving, ensuring efficient training time. Experiments were conducted at 10, 20, and 50 epochs to establish a baseline, with convergence generally achieved around 20 epochs, though more epochs were sometimes required for complex feature extraction tasks.

Experiment XXII*: *Batch Size
Batch size influences both memory usage and the stability of model training. Smaller batch sizes provide regular updates and can reduce overfitting but can lead to noisier gradients, whereas larger batch sizes stabilize gradient calculations but require more computational resources. Various batch sizes (e.g., 8, 16, 32, and 64) were tested to find a balance between memory constraints and training stability. Smaller batch sizes (e.g., 8 or 16) showed improvements in model generalization, especially when combined with dropout regularization, but required more training steps and incurred increased computational costs. Larger batch sizes stabilized training but were computationally demanding, with diminishing returns on model performance. A batch size of 16 provided an effective balance between training efficiency and model performance.

Experiment XXIII*: *Learning Rate
The learning rate controls the step size of each update during training. A learning rate too high risks overshooting optimal weights, while one too low may cause the model to converge slowly or get stuck in suboptimal local minima. Various learning rates (e.g., 0.001, 0.0001, and 0.00001) were tested using constant rates and schedules like cosine annealing and exponential decay to explore their impact on convergence speed and final model accuracy. Cosine annealing provided a smooth decline in the learning rate, which aided in the stability of convergence in the fusion model experiments. Additionally, using warm restarts in the learning rate helped avoid getting trapped in suboptimal minima, proving beneficial in achieving higher accuracy in PAD prediction.

5.4.2 Dimensionality Configuration: Single vs. Multi-Dimensional Models for PAD Prediction

In the task of PAD (Pleasure, Arousal, Dominance) affect prediction, a key decision involved determining the best training approach: should we develop and train a dedicated model for each modality separately or use a single model that can predict all three modalities simultaneously? The choice between these two approaches has implications for model complexity, compute efficiency, training and prediction accuracies.

Experiment XXIV*: *Training Individual Models for Each Modality
In this experiment, a separate model was trained for each of the three dimensions of PAD (Pleasure, Arousal, Dominance), treating each modality as an independent prediction task. Each model, using the same architecture, was fine-tuned specifically to optimize for one modality, allowing for more focused training. This approach assessed if a dedicated model per modality could lead to improved prediction accuracy by tailoring the model's learned representations to the unique characteristics of each PAD dimension.

Experiment XXV*: *Training a Unified Model to Predict All Three Modalities Simultaneously
The second experiment involved training a single model that could simultaneously predict Pleasure, Arousal, and Dominance values, as shown in Figure 4.6. This unified

model was designed with three prediction heads, each responsible for one of the PAD dimensions, allowing it to make simultaneous predictions for all three modalities and the access if the interdependencies between the dimensions helped in prediction.

Chapter 6
Results

The following chapter delves into the rationale behind the design choices, illustrating how each experiment contributed to the model's final implementation to predict Pleasure, Arousal, and Dominance values accurately in a multimodal context.

6.1 Video Model

In the video model, we explored various preprocessing steps to optimize the model's performance beginning with frame rate selection. With the DEAP dataset at 50 FPS and MITHOS at 25 FPS, including all frames at the original frame rates risked overwhelming the model due to the excessive input dimension relative to our output targets, making it difficult to recognize patterns effectively given the dataset sizes. After experimenting with various frame rates, we found that standardizing to 1 FPS significantly reduced input complexity, enabling the model to better capture essential affective features.

To identify the most relevant features from a frame for machine learning predictions, we experimented with various elements, including cropped faces, facial landmarks, cropped bodies, pose estimates, and combinations of these features, as each provides unique insights into affective cues. Firstly, body-based features are only available in the MITHOS dataset as the DEAP only has frontal face recordings. Secondly, the variations in body positioning among MITHOS participants, as shown in Fig. 6.1, highlight the challenges in using body-based features for affect prediction. This was because participants exhibited inconsistent poses, with some having their lower bodies cropped out due to recording errors, some holding objects like tablets, or some having their hands behind their backs throughout the experiment. These inconsistencies introduce noise into the data, making it difficult for the model to identify reliable patterns. Such variability, combined with limited body data availability, further reinforced our decision to exclude body features from our experiments and focus instead on facial features for a more standardized and effective approach.

Experiments with a baseline CNN model showed that using facial landmarks resulted in better accuracy than cropped faces, as the reduced input size was easier for the model to learn from. However, when using pretrained models, cropped face data outperformed

Figure 6.1: Inconsistencies in body data captured in participants in the MITHOS dataset

Video Model Selection				
Model	**Train MAE**	**Train PCC**	**Test MAE**	**Test PCC**
CNN (baseline)	1.6	0.6	2.9	0.04
Transformer	1.5	0.5	2.5	0.09
Pretrained ResNet	1	0.8	1.8	0.3
Pretrained ViViT	**0.9**	0.8	**1.10**	**0.30**
Pretrained SwinTransformer	1.2	0.8	1.4	0.24

Table 6.1: Performance comparison of different video models

landmarks. This was because these models were already pretrained on large datasets and captured low-level features, allowing them to focus on learning high-level, dataset-specific patterns in our use case. Thus, we proceeded with cropped face data as our primary input.

For time segmentation, we used the entire video for the DEAP dataset, as it consistently captured emotions. However, the MITHOS dataset required a more targeted approach as there were 6 different events(cues) in a single video (as described in Experiments VI to VIII). The sliding window strategy that considers 'N' seconds before and after the cue's timestamp, failed to capture emotions accurately, as the windows did not align well with emotional changes. Instead, we found that using the VSM and SSC logs provided more precise segments as they marked the onset of emotional changes effectively, starting right when the study master observed a shift. However, there were overlapping segments for different cues of the same participant, which created conflicting outputs for the same overlapping time frame, which added noise to the model's learning process. This was noticed as the training accuracies weren't increasing after a certain point. To overcome this, the time segments without overlaps provided clearer, non-conflicting data that improved the model's ability to learn affective patterns accurately, thus increasing training accuracies. Therefore, these time segments from the VSM and SSC logs without overlaps were considered.

Once the input features were defined, we applied them across various model architectures to determine the best approach for affective prediction. We experimented with several models: a CNN model, a custom transformer, a pretrained transformer, a pre-

trained ViViT, and a pretrained Swin Transformer. Each model's performance is outlined in the results Table 6.1, providing insights into their effectiveness in capturing and predicting emotional cues based on the input features. The evaluation was conducted by averaging the scores across all dimensions (PAD) to identify the model with the most accurate predictions, aiming for low Mean Absolute Error (MAE) scores and high Pearson Correlation Coefficient (PCC) scores.

The 3D CNN model was used as the baseline for video data, as CNN-based models are widely applied in affective analysis to capture spatial and temporal features. This model achieved good training accuracy but had high errors, with a Test MAE of 2.9, and a relatively low PCC of 0.04, indicating limitations in generalizing temporal patterns effectively. The custom Transformer model performed better than the CNN, with a test MAE of 2.5 and PCC of 0.09. This improvement can be attributed to the Transformer's attention mechanism combined with MHSA and FFN, which help capture spatial and temporal dependencies. However, custom Transformers require extensive data to fully learn intricate patterns, which is limited in our use case. This led us to experiment with pretrained models.

The pretrained ResNet (CNN-based) model showed promising results, achieving a lower test MAE of 1.8, and PCC of 0.3, highlighting the benefit of using pretrained models to overcome data limitations. As transformer-based models have proved to outperform CNN-based models in various domains, we then explored pretrained Transformer models, finding that the pretrained ViViT model outperformed all previous models, achieving the highest training metrics, lowest Test MAE of 1.1, and the best Test PCC of 0.3, indicating effective generalization across spatial and temporal features. Lastly, we tested the pretrained SwinTransformer, which, while pretrained on facial data and potentially suitable for our task, underperformed with a Test MAE of 1.4 and PCC of 0.24. The SwinTransformer's core mechanism of aggregating patches within frames, as shown in Fig. 5.5, limits its ability to capture global spatial patterns, making it less effective than the ViViT model for this use case. As an initial analysis, the pretrained ViViT model proved to be effective and therefore, we consider this as the video model.

After finalizing the model, we conducted an ablation study with the pretrained ViViT model across three configurations: training on the DEAP dataset alone, on the MITHOS dataset alone, and a combined approach of pretraining on DEAP with transfer learning on MITHOS, as shown in Table 6.2. The goal was to evaluate the model's performance and validate the benefits of fine-tuning with transfer learning. Pretraining on DEAP allows the model to capture general features, creating a foundational context that is particularly useful when data is limited in the target domain. As seen in the results, direct training on MITHOS led to poorer performance, likely due to the lack of contextual reference and limited data for a complex model like ViViT. However, the setup with pretraining on DEAP and fine-tuning on MITHOS achieved the best performance, with an overall Test MAE of 1.1 and Test PCC of 0.3 as well as a better Range-2 Accuracy of 72% and Class Accuracy of 60%. This configuration highlights the advantage of transfer learning, as leveraging knowledge from a related dataset significantly enhances the model's adaptability and effectiveness on new data.

Given the three dimensions (Pleasure, Arousal, and Dominance), we sought an approach that would compute the models more efficiently, ideally leveraging shared information across dimensions when available. To do this we compared the performance of using separate models for each dimension versus a single model for all dimensions as shown in Table 6.3. The results of the individual models demonstrated higher training accuracy metrics, indicating a better fit to the individual dimensions but suggesting borderline

Ablation Study							
Model	**Metric**	**Train MAE**	**Train PCC**	**Test MAE**	**Test PCC**	**Test Range-2 Accuracy**	**Test Class Accuracy**
DEAP	Pleasure	2.6	0.6	2.1	0	61	54
	Arousal	2.65	0.4	1.6	0.2	74	58
	Dominance	1.9	0.6	1.8	0.2	67	43
	Overall	**2.4**	**0.5**	**1.8**	**0.1**	**67.3**	**51.7**
MITHOS	Pleasure	3	0.51	1.722	0	50.63	43.2
	Arousal	3.127	0.348	1.344	0.166	61.42	55.1
	Dominance	2.337	0.492	1.53	0.1	55.61	34.4
	Overall	**2.8**	**0.5**	**1.5**	**0.1**	**55.9**	**44.2**
DEAP+ MITHOS	Pleasure	1	0.6	1.3	0.4	68	60
	Arousal	0.9	0.9	0.9	0.12	78	63
	Dominance	1	0.8	1.1	0.45	70	57
	Overall	**1.0**	**0.8**	**1.1**	**0.3**	**72**	**60**

Table 6.2: Ablation Study Results for DEAP, MITHOS, and DEAP+MITHOS datasets on the Video model

overfitting, as they did not generalize as well as the unified model. In contrast, the unified model with shared parameters across all dimensions showed slightly lower training accuracies yet generalized slightly more effectively, as evidenced by improved test performance. Notably, the test accuracies for each PAD dimension were individually better in the unified model compared to the separate models, resulting in better overall performance by capturing inter-dimensional relationships and reducing overfitting. Emotions often display patterns and cues in one dimension that inform or enhance predictions in the others, with correlations across the PAD dimensions leveraged effectively through shared parameters in the model. Individual models also come with an added overhead for maintaining 3 separate models instead of one. Additionally, using one model reduces redundancy, as shared features are learned together rather than separately for each dimension, leading to a more efficient and cohesive learning process making more accurate predictions across PAD values on the test data.

Individual Models vs One Model					
Architecture	**Model**	**Train MAE**	**Train PCC**	**Test MAE**	**Test PCC**
Individual Model	Pleasure	0.8	0.8	1.15	0.5
	Arousal	0.9	0.9	0.9	0.4
	Dominance	0.7	0.9	1.4	0.1
	Overall	**0.8**	**0.87**	**1.15**	**0.3**
Combined Model	Pleasure	1	0.6	1.3	0.4
	Arousal	0.9	0.9	0.9	0.12
	Dominance	1	0.8	1.1	0.45
	Overall	**1.0**	**0.8**	**1.1**	**0.3**

Table 6.3: Comparison of Individual and Combined Models based on various metrics.

The pretrained ViViT model consists of 11 encoder layers, which we aimed to transfer-learn on DEAP to capture general affective features and subsequently fine-tune on

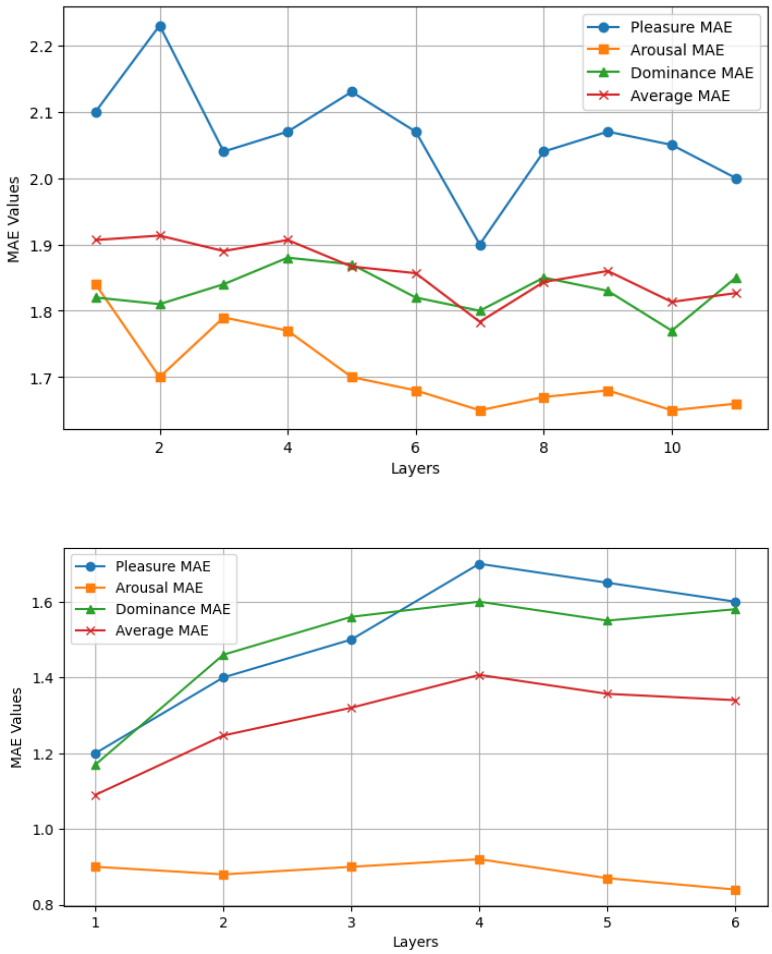


Figure 6.2: Mean Absolute Error (MAE) across ViViT's trainable layers for PAD pretrained on DEAP (top) and further fine-tuned on MITHOS (bottom). Layer numbers start from the final layers of the model i.e. Layer 1 corresponds to the last layer, Layer 2 corresponds to the last 2 layers, and so on.

MITHOS for dataset-specific features. Training all layers of the model can prove to be counterproductive, as it risks overwriting generalized video representations with data-specific patterns, leading to overfitting. To balance this, we need to freeze the weights of the earlier layers(that capture general low-level data), updating only the latter layers. We experimented by progressively unfreezing layers from the last layer backward to identify the optimal configuration for accuracy. The graph in Fig. 6.2 (top) revealed that training up to the 7th layer provided the best results while unfreezing additional layers degraded performance by embedding DEAP-specific details at the expense of generalizability. Consequently, we froze the first 4 layers and trained only the last 7 layers on DEAP.

We then applied the same strategy to further fine-tune the model on the MITHOS dataset, focusing on use-case-specific adaptation. As seen in Fig. 6.2 (bottom) graph, train-

ing only the final layer yielded optimal accuracy, indicating that most general features learned from DEAP aligned well with MITHOS requirements. This approach—training deeper layers on DEAP for generalization and the final layer on MITHOS for specificity—optimized the model for accurate PAD predictions across both datasets.

In conclusion, the optimized VideoViT model, with strategically fine-tuned layers, effectively captures both general and dataset-specific affective features, making it a robust feature extractor for our fusion model. By leveraging the DEAP dataset for foundational affective cues and the MITHOS dataset for use-case-specific nuances, this model is well-suited to provide high-quality, contextually adaptive video features that enhance the overall multimodal fusion model's capability for accurate PAD predictions.

6.2 Audio Model

For the audio model, we use only the MITHOS dataset as the DEAP doesn't have audio signals. For the preprocessing of MITHOS time segments, the non-overlapping time segments yield better performance, as established in the previous section which logically aligns with the need for clear, non-conflicting data segments, allowing the model to learn more effectively from distinct emotional transitions. Therefore, we proceeded with non-overlapping timestamps from the VSM and SSC logs to enhance the model's accuracy in affect prediction.

Audio Model Selection							
Model	**Dimension**	**Train MAE**	**Train PCC**	**Test MAE**	**Test PCC**	**Test Range-2 Accuracy**	**Test Class Accuracy**
3DCNN (baseline)	Pleasure	0.8	0.6	1.9	-0.08	45	38
	Arousal	0.6	0.7	1.8	0.1	50	45
	Dominance	1.1	0.5	2.1	0.2	53	43
	Overall	**0.8**	**0.6**	**1.8**	**0.1**	**49.3**	**42.0**
Wav2Vec without fine-tuning	Pleasure	-	-	1.6	-0.5	70	43
	Arousal	-	-	1.5	0.1	80	48
	Dominance	-	-	1.5	0.4	69	45
	Overall	-	-	**1.5**	**0.1**	**73**	**45.3**
Wav2Vec with fine-tuning	Pleasure	0.4	0.7	1.4	-0.1	80	45
	Arousal	0.2	0.8	0.9	-0.4	82	50
	Dominance	0.3	0.7	1.4	0.2	71	50
	Overall	**0.3**	**0.7**	**1.2**	**0.2**	**77.7**	**48.3**

Table 6.4: Performance comparison of different audio models

For the audio model, whose results are represented in Table 6.4, we began by using a baseline CNN model as a reference for performance. However, Wav2Vec2, a model pre-trained on extensive audio data, emerged as a more suitable choice given its robust feature extraction capabilities for affective audio analysis. The pretraining on large datasets made Wav2Vec2 an ideal candidate for our use case, as it captures foundational audio representations well-suited to identifying affective cues. To gain initial insights, we assessed Wav2Vec2's prediction accuracies without additional training, which provided a valuable benchmark and highlighted the model's intrinsic ability to recognize general audio features. The evaluation of Wav2Vec2 without fine-tuning showed promising

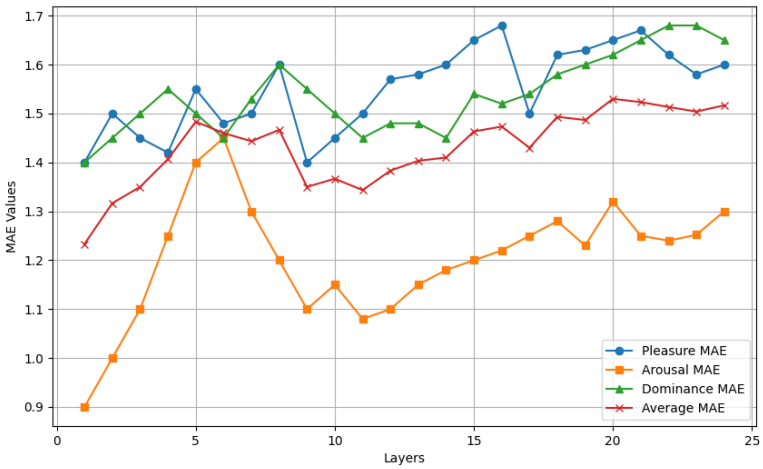


Figure 6.3: Mean Absolute Error (MAE) across Wav2Vec2's trainable layers for PAD fine-tuned on MITHOS. Layer numbers start from the final layers of the model i.e. Layer 1 corresponds to the last layer, Layer 2 corresponds to the last 2 layers, and so on.

results, achieving a Test MAE of 1.5 across dimensions and a notable improvement in Test Range-2 Accuracy, averaging 73%. This indicated that Wav2Vec2's pretraining enabled it to capture general audio-based affective cues with relatively high accuracy, even without further training.

To adapt Wav2Vec2 more specifically to our context, we fine-tuned only the final layers, focusing on the MITHOS dataset as the DEAP dataset lacks audio signals. This targeted fine-tuning allowed Wav2Vec2 to learn high-level, dataset-specific affective patterns aligned with our use case while preserving its foundational audio representations. After fine-tuning, we observed further improvements in accuracy and consistency across dimensions, with the Test MAE lowering to 1.2 overall and Test PCC increasing to 0.2. Additionally, Test Range-2 Accuracy improved to 77.7%, confirming that this fine-tuning approach enabled Wav2Vec2 to balance its pre-trained foundational features with specific cues relevant to PAD prediction in our context. This approach allowed for efficient learning of relevant emotional cues for PAD prediction, balancing specificity to our dataset with generalizability across similar affective contexts.

The Wav2Vec2 model is designed to predict all three PAD dimensions simultaneously using shared weights, allowing it to leverage inter-dimensional relationships. Consequently, we did not experiment with separate models for each dimension, as the shared-weight approach is more efficient and effectively captures correlations across Pleasure, Arousal, and Dominance.

The Wav2Vec2 model typically has 24 transformer layers, which are trainable and responsible for capturing high-level audio representations. In fine-tuning scenarios, however, it's common to train only the final few layers, to balance computational efficiency and model performance. In our case, training only the final layer of the Wav2Vec2 model proved to be effective as shown in Fig. 6.3, as it provided optimal results. This approach allowed the model to adapt to the dataset-specific affective patterns in MITHOS without compromising the foundational audio representations learned during pretraining,

achieving a balance between efficiency and accuracy.

In conclusion, the fine-tuned Wav2Vec2 model, with its ability to effectively capture both foundational and dataset-specific affective audio features, is well-suited as a feature extractor for the fusion model. This configuration enables the model to provide high-quality audio feature representation that complements other modalities in the fusion model, enhancing the overall effectiveness of multimodal affect analysis.

6.3 Fusion Model

In evaluating the fusion model, we focused exclusively on the MITHOS dataset, as it is the only dataset available with both audio and video inputs—critical components for our multimodal approach. The DEAP dataset, which lacks audio signals, was therefore unsuitable for these fusion experiments. For the fusion model, no additional preprocessing is required, as we directly use the feature representations from the final layers of the audio and video models as inputs. The final layers are typically chosen for fusion because they provide high-level, abstract representations that capture the essential affective patterns learned by each modality. These features are well-suited for combining in a fusion model, as they distill the most relevant information for predicting PAD values, reducing noise from lower-level features. They also prove to be better than the raw inputs as these feature representations are more context-aware considering the given inputs.

Experiments involving more than the final layers introduced excessive complexity in the fusion model's input, making it harder for the model to learn meaningful relationships between modalities given the heightened dimensionality and small data size. Including additional layers increased the dimensionality of the input, leading to greater computational overhead and potential overfitting. Furthermore, lower-level features often capture modality-specific nuances, which can introduce redundancy and conflict when combined across modalities. By focusing on the final layers, we ensure that the fusion model receives concise, complementary representations, improving its predictive accuracy and efficiency. The results of the fusion model are illustrated in the Table 6.5

Now, we experiment with Early Fusion, Cross-Modal Fusion, an RNN-based FFN Fusion model, and a Transformer model. Early Fusion is tested to confirm that simply concatenating audio and video inputs falls short in producing optimal predictions, as it lacks the ability to model interactions between modalities. This limitation restricts its capacity to capture the nuanced dependencies essential for accurate affect analysis, underscoring the need for a more sophisticated approach that can effectively represent inter-modal relationships. We use features extracted from the audio and video models as inputs for the next set of models.

The results from the table indicate that while the Cross-Modal Attention model offers improvement over Early Fusion, as seen in Table 6.5, by allowing basic alignment between audio and video features and highlighting the most relevant parts of the input. It still falls short in effectively capturing time-based dependencies essential for understanding dynamic emotional states. This limitation is evident in the inconsistent performance across dimensions, particularly in Dominance, where the Test PCC remains low (0.1) and Test MAE is relatively high (1.5). Cross-modal attention focuses primarily on pairwise relationships at a single moment, rather than capturing how these relationships evolve over time. This narrow focus restricts its ability to generalize to varying temporal patterns, which is necessary for interpreting complex and shifting emotional cues. Consequently,

Fusion Model Selection							
Model	**Dimension**	**Train MAE**	**Train PCC**	**Test MAE**	**Test PCC**	**Test Range-2 Accuracy**	**Test Class Accuracy**
Early Fusion Model	Pleasure	1.3	0.4	1.8	0.2	60	58
	Arousal	1.1	0.2	1.2	-0.3	85	80
	Dominance	1.2	0.1	1.7	0.1	70	65
	Overall	**1.2**	**0.2**	**1.6**	**0.0**	**71.7**	**67.7**
Cross Modal Attention (baseline)	Pleasure	1.1	0.4	1.1	0.2	68	60
	Arousal	1.3	0.1	0.6	0.0	92	90
	Dominance	1.4	0.1	1.5	0.1	60	63
	Overall	**1.3**	**0.2**	**1.1**	**0.1**	**73.3**	**71.0**
RNN-LSTM	Pleasure	1.6	0.06	1.2	0.2	62	60
	Arousal	1.3	0.2	0.9	0.0	92	85
	Dominance	1.4	0.0	1.5	0.0	71	59
	Overall	**1.4**	**0.09**	**1.2**	**0.07**	**75**	**68**
Transformer	Pleasure	0.9	0.3	1.8	0.3	73	64
	Arousal	0.7	0.5	0.8	0.5	95	92
	Dominance	1.2	0.25	1.5	0.25	75	65
	Overall	**0.9**	**0.4**	**1.0**	**0.4**	**81.0**	**73.7**

Table 6.5: Performance comparison of different fusion models

although Cross-Modal Attention allows for some inter-modal interaction, its limited capacity to model temporal dependencies results in suboptimal performance for affect analysis, underscoring the need for more advanced methods capable of handling time-based interdependencies across modalities.

The RNN-based FFN Fusion Model, incorporates an LSTM layer to introduce temporal modeling, which aided in capturing sequence dependencies to address some shortcomings of the Cross-Modal Attention model. The RNN-based model struggled with longer sequences, facing challenges with vanishing gradients and limited parallel processing capabilities, which hindered efficiency and scalability. Additionally, small batch sizes were required for computational feasibility, further impacting the model's overall performance. As shown in the table 6.5, the RNN-based FFN Fusion model does not outperform the Cross-Modal Attention model. These findings indicate that more advanced attention-based models, such as transformers, are necessary to better capture nuanced interactions and dependencies across modalities and time.

The Transformer model emerged as the most effective choice for fusion, utilizing self-attention mechanisms to capture both spatial and temporal dependencies across modalities. Unlike previous models, the Transformer's multi-head self-attention allows it to identify complex, context-rich relationships within audio-video data, leading to significant improvements in PAD prediction accuracy. As shown in the table, the Transformer model achieved the lowest overall Test MAE (1.0) and the highest Test PCC (0.4), indicating a robust ability to generalize across emotional dimensions. The Transformer's parallel processing capability also enables efficient training on large datasets, overcoming computational limitations faced by RNNs. These attributes make the Transformer model the most suitable for accurately integrating and interpreting audio and video inputs,

Ablation Study Fusion Model							
Architecture	**Model**	**Train MAE**	**Train PCC**	**Test MAE**	**Test PCC**	**Test Range-2 Accuracy**	**Test Class Accuracy**
Individual Model	Pleasure	1.1	0.6	1.3	0.1	66	60
	Arousal	0.8	0.7	0.7	0.4	95	90
	Dominance	1.1	0.5	1.5	0.1	74	60
	Overall	**1.0**	**0.6**	**1.2**	**0.2**	**78.3**	**70**
Combined Model	Pleasure	0.9	0.3	1	0.3	73	64
	Arousal	0.7	0.5	0.8	0.5	95	92
	Dominance	1.1	0.4	1.3	0.25	75	65
	Overall	**0.9**	**0.4**	**1.0**	**0.4**	**81.0**	**73.7**

Table 6.6: Performance comparison of individual model architectures and combined model architecture (shared parameters) for fusion model

providing a cohesive and comprehensive affect analysis across the PAD dimensions.

The Transformer fusion model is not pretrained, so to optimize its performance for our specific use case, we fine-tuned all available parameters to align closely with the MITHOS dataset. This comprehensive tuning allowed the model to effectively learn dataset-specific affective patterns, ensuring that it accurately captures and integrates the nuanced emotional cues required for PAD prediction.

The ablation study compares two fusion model approaches explained in Table 6.6: individual models for each PAD dimension and a combined model that predicts all three dimensions simultaneously. The combined model generally performs better, with a slight advantage in test PCC (0.4 overall) compared to the individual models (0.2 overall). Although the individual models achieve slightly better MAE for specific dimensions, such as Arousal (0.7 vs. 0.8), the combined model offers a balanced performance across all metrics, especially with higher Test Class Accuracy (73.7% vs. 70%).

The combined model's improved efficiency lies in its ability to leverage shared features across dimensions, which reduces the redundancy of training and managing three separate models. This unified approach simplifies computational demands, requiring fewer resources and training time, while still capturing the interdependencies among PAD dimensions. Additionally, the combined model benefits from shared learned patterns, which enhances generalization and produces more stable predictions across different dimensions. Overall, the combined model is both more computationally efficient and effective for predicting PAD values in the fusion framework.

6.4 Overall Evaluation

The table 6.7 provides an overall summary of model performance across video, audio, and fusion models for PAD prediction. Each modality demonstrates strong predictive capabilities, especially considering the non-intrusive nature of audio and video inputs. However, the Fusion model exhibits a marginally superior performance compared to the individual audio and video models, highlighting the effectiveness of combining multimodal features.

Overall Accuracy							
Model	**Dimension**	**Train MAE**	**Train PCC**	**Test MAE**	**Test PCC**	**Test Range-2 Accuracy**	**Test Class Accuracy**
Video Model	Pleasure	1	0.6	1.3	0.4	68	60
	Arousal	0.9	0.9	0.9	0.12	78	63
	Dominance	1	0.8	1.1	0.45	70	57
	Overall	**1.0**	**0.8**	**1.1**	**0.3**	**72**	**60**
Audio Model	Pleasure	0.4	0.7	1.4	-0.1	80	45
	Arousal	0.2	0.8	0.9	-0.4	83	50
	Dominance	0.3	0.7	1.4	0.2	71	50
	Overall	**0.3**	**0.7**	**1.2**	**0.2**	**78**	**48.3**
Fusion Model	Pleasure	0.9	0.3	1.0	0.3	73	64
	Arousal	0.7	0.5	0.8	0.5	95	92
	Dominance	1.1	0.4	1.3	0.25	75	65
	Overall	**0.9**	**0.4**	**1.0**	**0.4**	**81.0**	**73.7**

Table 6.7: Comparison of performance on Video, Audio, and Fusion Models

In evaluating the model, Mean Absolute Error (MAE) was chosen over Mean Squared Error (MSE) as a primary metric. MAE measures the average absolute difference between predicted and actual values, providing a straightforward interpretation of prediction accuracy without penalizing larger errors, which MSE tends to do. This makes MAE more suitable for assessing the model's performance, especially in PAD prediction, where capturing subtle differences across dimensions is essential. The transformer-based Fusion model achieves a slightly lower overall Test MAE and a higher Test PCC compared to the standalone models, indicating a stronger correlation with actual PAD values and more accurate generalization. This metric validates the fusion model's consistency and robustness across different emotional dimensions. An important metric, Range-2 Accuracy, measures how often the predicted values fall within a +/-2 range of the actual values. This is chosen because the MAE values for all models lie between 1 and 2. This range accuracy indicates that the fusion model's outputs lie within a smaller range of the actual values as compared the the audio/video models, and therefore generalizes well across diverse conditions (different cues), instead of simply targeting the mean outcome ranges. Lastly, Class-based Accuracy provides a means to compare results with other state-of-the-art methods using different modalities or approaches. By categorizing predictions, this metric enables direct performance comparisons, underscoring the effectiveness of the fusion approach relative to existing models in affective analysis. Class-based Accuracy scores are often lower than Range-2 Accuracy due to the rigid class boundaries; for example, a prediction of 4.5 instead of 4 is marked incorrect, even though it is close to the target.

These results confirm that leveraging the complementary strengths of both modalities, provides a more comprehensive and reliable solution for PAD prediction, making it an effective approach for affective analysis in multimodal contexts.

Chapter 7
Conclusion and Discussion

7.1 Conclusion

The following inferences demonstrate how this research enhances human-virtual agent interactions, enabling virtual agents to respond in a more natural and empathetic manner by leveraging audio and video data for affective cues. To encapsulate the emotional spectrum, this research aimed to predict Pleasure, Arousal, and Dominance (PAD) values using non-intrusive multimodal data (audio and video) which can be integrated into virtual agents to enable real-time affect analysis in human-virtual agent interactions. Traditional models often overlook Dominance, yet it plays a vital role in distinguishing emotions like anger and fear, which share high arousal but differ in the sense of control. By incorporating Dominance alongside Pleasure and Arousal, this work seeks to provide a fuller, context-aware understanding of emotions, allowing virtual agents to respond more effectively and empathetically in interactions.

The DEAP dataset provided a good starting point for affect analysis, capturing general affective patterns, however, its setup lacked the dyadic interaction between a human and a virtual agent that is central to our use case. In contrast, the MITHOS dataset aligns closely with our objectives with enriched data from non-intrusive modalities (audio and video) along with face and body data, offering a more relevant context that supports fine-tuning within human-virtual agent interactions.

Feature extraction was performed through pretrained models—Wav2Vec2 for audio data and ViViT for video data—allowing each modality to capture nuanced, modality-specific patterns. The extracted features were then integrated using a model-based fusion approach. Unlike early or basic late fusion methods, model-based fusion enabled the model to dynamically weigh the relevance of each modality within the context of the interaction, capturing interdependencies that are critical for a comprehensive affective analysis. This adaptive fusion strategy improves prediction accuracy across PAD dimensions and offers flexibility for future multimodal expansions, as additional data types could be seamlessly integrated, enhancing the model's adaptability and robustness in varied real-world applications.

In this research, the fused model demonstrated marginally better performance compared to individual audio and video models, showcasing the benefits of combining modalities for PAD prediction. Both the audio (Wav2Vec2) and video (ViViT) models are heavily pretrained on large and diverse datasets, which contributes significantly to their robustness and individual accuracy. The fusion model, however, is trained on a relatively small dataset, yet it still manages to slightly outperform the individual models in predicting PAD values, reflecting enhanced robustness and accuracy through our fusion approach. This marginal improvement suggests that the fusion model has the potential to leverage complementary information from both audio and video inputs effectively. With a larger dataset or additional fine-tuning on data specific to similar human-virtual interaction use cases, like an expanded MITHOS dataset, the fusion model could achieve even stronger performance. While this research indicates potential in the fusion approach, its ultimate effectiveness could be better realized with more extensive training data.

Interestingly, Arousal performed slightly better in an individual model, suggesting it has less dependency on Pleasure and Dominance. This observation aligns with the confusion matrix analysis shown in Fig. 3.4 and Fig. 3.11, which indicates low interdependency of Arousal with the other two dimensions, showing that Arousal is more independent compared to the other dimensions. This insight reflects the unique relationships between the PAD dimensions, highlighting the value of a combined model while recognizing Arousal's distinct predictive behavior.

Our results show that each modality contributes differently to predicting PAD dimensions within the fusion model. The audio modality plays a larger role in predicting Pleasure and Arousal, likely due to vocal cues like tone, pitch, and intensity, which convey emotions such as excitement or calmness. For instance, a higher pitch and increased speech rate often correlate with heightened arousal, while a warm and melodic tone may indicate pleasure. These nuances in vocal expression make audio data particularly effective for capturing the subtle fluctuations in Pleasure and Arousal dimensions. Conversely, the video modality is more influential in predicting Dominance, as visual, non-verbal cues—such as facial expressions or gestures strongly indicate confidence and control, key aspects of dominance. This insight underscores the potential of the fusion model to adaptively weight each modality based on the dimension being predicted, improving overall accuracy. Building on these findings, when applied to virtual agents, the relevant features in the audio (like tone, pitch, or intensity) and video (like facial expressions) can be adjusted to regulate the PAD values in the agent, enabling it to display desired emotional responses.

By incorporating PAD-based affect predictions, virtual agents can more accurately interpret and adapt to users' emotional cues, creating personalized interactions that boost engagement, empathy, and rapport. This model establishes a foundation for virtual agents to deliver contextually relevant and emotionally attuned responses, ultimately enriching the overall user experience.

7.2 Discussion

While this research achieved significant strides in multimodal affect analysis, several limitations remain. Dataset size constraints were a challenge, limiting the model's ability to capture the full range of nuanced emotional states. Additionally, relying solely on non-intrusive modalities like audio and video, while practical, may not capture subtle emotional cues as effectively as physiological data. Future research could address these

limitations by incorporating additional modalities, such as text, which could further enrich the model's emotional context and improve accuracy.

Transfer learning proved valuable in this study, enabling the model to leverage knowledge from pretrained datasets and adapt effectively to the limited data available. This approach could be extended to similar tasks with transfer learning, using the fusion model as a foundation to improve performance with more relevant data.

Currently, the model outputs PAD values on a 1–9 scale, but ideally, these values should be normalized to a range of -1 to 1 for consistency with the PAD model's theoretical framework. Adapting this normalization based on the use case would provide more flexibility in future implementations, aligning with standards in affective computing and enhancing interpretability across varied contexts.

In comparing the final results with previous research, our audio-video fusion model demonstrates competitive performance relative to state-of-the-art methods that rely on various modalities. Studies that have used more intrusive data sources, such as EEG or physiological signals, often achieve higher accuracies due to the rich emotional cues provided by these modalities. However, our model's results show that with careful design, non-intrusive audio and video data can achieve comparable accuracy, providing a practical and accessible alternative for affect analysis. Our fusion model, through dynamic weighting and adaptive feature extraction, effectively balances the strengths of audio and video modalities. While EEG or ECG-based approaches may capture subtler variations in emotional states, our approach holds significant advantages in real-time applications and user accessibility. Overall, our results affirm that combining audio and video data provides a viable and robust solution for PAD prediction, standing well against other approaches that require more complex or intrusive data sources.

The findings from this research offer valuable applications in enhancing virtual agent technology, particularly in settings requiring dynamic emotional responsiveness, such as education, and social coaching as done in the MITHOS system. By effectively predicting PAD values from non-intrusive audio and video data, our model enables virtual agents to interpret and adapt to user emotions in real-time, allowing for more natural, context-aware interactions. This approach provides a practical alternative to sensor-dependent affect analysis, balancing accuracy and accessibility, and paving the way for virtual agents to deliver emotionally attuned responses. Such advancements can significantly improve user experience and engagement, positioning virtual agents as empathetic and responsive partners in human-centered applications.

Bibliography

[1] Salma Alhagry, Aly Aly Fahmy, and Reda A El-Khoribi. 2017. Emotion recognition based on EEG using LSTM recurrent neural network. *International Journal of Advanced Computer Science and Applications* 8, 10 (2017).

[2] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. 2021. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*. 6836–6846.

[3] Iris Bakker, Theo Van Der Voordt, Peter Vink, and Jan De Boon. 2014. Pleasure, arousal, dominance: Mehrabian and Russell revisited. *Current Psychology* 33 (2014), 405–421.

[4] Rukshan Batuwita and Vasile Palade. 2013. Class imbalance learning methods for support vector machines. *Imbalanced learning: Foundations, algorithms, and applications* (2013), 83–99.

[5] Chirag Bhuvaneshwara, Manuel Anglet, Bernhard Hilpert, Lara Chehayeb, Ann-Kristin Meyer, Daksitha Withanage Don, Dimitra Tsovaltzi, Patrick Gebhard, Antje Biermann, Sinah Auchtor, and others. 2023. MITHOS-Mixed Reality Interactive Teacher Training System for Conflict Situations at School. In *ISLS Annual Meeting 2023*. 42.

[6] Hessel R Bosma. 2022. *Audio-visual Correlation from Cross-modal Attention in Self-supervised Transformers on Videos of Musical Performances*. Master's thesis. University of Twente.

[7] Margaret M Bradley and Peter J Lang. 1994. Measuring emotion: the self-assessment manikin and the semantic differential. *Journal of behavior therapy and experimental psychiatry* 25, 1 (1994), 49–59.

[8] Joost Broekens, Anne Pronker, and Marian Neuteboom. 2010. Real time labeling of affect in music using the affectbutton. In *Proceedings of the 3rd international workshop on Affective interaction in natural environments*. 21–26.

[9] Aleksandra Cerekovic, Oya Aran, and Daniel Gatica-Perez. 2014. How do you like your virtual agent?: Human-agent interaction experience through nonverbal features and personality traits. In *Human Behavior Understanding: 5th International Workshop, HBU 2014, Zurich, Switzerland, September 12, 2014. Proceedings 5*. Springer, 1–15.

[10] Lara Chehayeb, Chirag Bhuvaneshwara, Manuel Anglet, Bernhard Hilpert, Ann-Kristin Meyer, Dimitra Tsovaltzi, Patrick Gebhard, Antje Biermann, Sinah Auchtor, Nils Lauinger, and others. 2024. MITHOS: Interactive Mixed Reality Training to Support Professional Socio-Emotional Interactions at Schools. *arXiv preprint arXiv:2409.12968* (2024).

[11] Shizhe Chen and Qin Jin. 2016. Multi-modal conditional attention fusion for dimensional emotion prediction. In *Proceedings of the 24th ACM international conference on Multimedia*. 571–575.

[12] Jadisha Cornejo and Helio Pedrini. 2019. Bimodal emotion recognition based on audio and facial parts using deep convolutional neural networks. In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*. IEEE, 111–117.

[13] Muhammad Najam Dar, Muhammad Usman Akram, Sajid Gul Khawaja, and Amit N Pujari. 2020. CNN and LSTM-based emotion charting using physiological signals. *Sensors* 20, 16 (2020), 4551.

[14] Paul Ekman. 1992. An argument for basic emotions. *Cognition & emotion* 6, 3-4 (1992), 169–200.

[15] Patrick Gebhard. 2005. ALMA: a layered model of affect. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*. 29–36.

[16] Asmaul Hosna, Ethel Merry, Jigmey Gyalmo, Zulfikar Alom, Zeyar Aung, and Mohammad Abdul Azim. 2022. Transfer learning: a friendly introduction. *Journal of Big Data* 9, 1 (2022), 102.

[17] Juhee Kim and Hyeon-Jeong Suk. 2020. Prediction of the Emotion Responses to Poster Designs based on Graphical Features: A Machine learning-driven Approach. *Archives of Design Research* 33, 2 (2020), 39–55. DOI:http://dx.doi.org/10.15187/adr.2020.05.33.2.39

[18] Sander Koelstra, Christian Muhl, Mohammad Soleymani, Jong-Seok Lee, Ashkan Yazdani, Touradj Ebrahimi, Thierry Pun, Anton Nijholt, and Ioannis Patras. 2011. Deap: A database for emotion analysis; using physiological signals. *IEEE transactions on affective computing* 3, 1 (2011), 18–31.

[19] Jean Kossaifi, Georgios Tzimiropoulos, Sinisa Todorovic, and Maja Pantic. 2017. AFEW-VA database for valence and arousal estimation in-the-wild. *Image and Vision Computing* 65 (2017), 23–36.

[20] Jean Kossaifi, Robert Walecki, Yannis Panagakis, Jie Shen, Maximilian Schmitt, Fabien Ringeval, Jing Han, Vedhas Pandit, Antoine Toisoul, Björn Schuller, and others. 2019. Sewa db: A rich database for audio-visual emotion and sentiment research in the wild. *IEEE transactions on pattern analysis and machine intelligence* 43, 3 (2019), 1022–1040.

[21] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436–444.

[22] Xiang Li, Dawei Song, Peng Zhang, Guangliang Yu, Yuexian Hou, and Bin Hu. 2016. Emotion recognition from multi-channel EEG data through convolutional recurrent neural network. In *2016 IEEE international conference on bioinformatics and biomedicine (BIBM)*. IEEE, 352–359.

[23] Hailun Lian, Cheng Lu, Sunan Li, Yan Zhao, Chuangao Tang, and Yuan Zong. 2023. A survey of deep learning-based multimodal emotion recognition: Speech, text, and face. *Entropy* 25, 10 (2023), 1440.

[24] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. 2022. Video swin transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 3202–3211.

[25] Albert Mehrabian. 1970. A semantic space for nonverbal behavior. *Journal of consulting and clinical Psychology* 35, 2 (1970), 248.

[26] Albert Mehrabian. 1980. Basic dimensions for a general psychological theory: Implications for personality, social, environmental, and developmental studies. (1980).

[27] Albert Mehrabian and James A Russell. 1974. *An approach to environmental psychology.* the MIT Press.

[28] Liyu Meng, Yuchen Liu, Xiaolong Liu, Zhaopei Huang, Wenqiang Jiang, Tenggan Zhang, Chuanhe Liu, and Qin Jin. 2022. Valence and arousal estimation based on multimodal temporal-aware features for videos in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2345–2352.

[29] Ali Mollahosseini, Behzad Hasani, and Mohammad H Mahoor. 2017. Affectnet: A database for facial expression, valence, and arousal computing in the wild. *IEEE Transactions on Affective Computing* 10, 1 (2017), 18–31.

[30] Durgesh Nandini, Jyoti Yadav, Asha Rani, and Vijander Singh. 2023. Design of subject independent 3D VAD emotion detection system using EEG signals and machine learning algorithms. *Biomedical Signal Processing and Control* 85 (2023), 104894. DOI: http://dx.doi.org/https://doi.org/10.1016/j.bspc.2023.104894

[31] Behnaz Nojavanasghari, Tadas Baltrušaitis, Charles E Hughes, and Louis-Philippe Morency. 2016. Emoreact: a multimodal approach and dataset for recognizing emotional responses in children. In *Proceedings of the 18th acm international conference on multimodal interaction*. 137–144.

[32] Yagya Raj Pandeya and Joonwhoan Lee. 2021. Deep learning-based late fusion of multimodal information for emotion classification of music video. *Multimedia Tools and Applications* 80 (2021), 2887–2905.

[33] L. Petrescu, C. Petrescu, A. Oprea, O. Mitruț, G. Moise, A. Moldoveanu, and F. Moldoveanu. 2021. Machine Learning Methods for Fear Classification Based on Physiological Features. *Sensors (Basel, Switzerland)* 21, 13 (2021), 4519. DOI:http://dx.doi.org/10.3390/s21134519

[34] Hubert Plisiecki and Aleksandra Sobieszek. 2024. Extrapolation of affective norms using transformer-based neural networks and its application to experimental stimuli selection. *Behavior Research Methods* 56 (2024), 4716–4731. DOI:http://dx.doi.org/10.3758/s13428-023-02212-3

[35] R Gnana Praveen, Patrick Cardinal, and Eric Granger. 2023. Audio-visual fusion for emotion recognition in the valence-arousal space using joint cross-attention. *IEEE Transactions on Biometrics, Behavior, and Identity Science* (2023).

[36] Luca Romeo, Andrea Cavallo, Lucia Pepa, Nadia Bianchi-Berthouze, and Massimiliano Pontil. 2019. Multiple instance learning for emotion recognition using physiological signals. *IEEE Transactions on Affective Computing* 13, 1 (2019), 389–407.

[37] Viktor Rozgić, Shiv N Vitaladevuni, and Rohit Prasad. 2013. Robust EEG emotion classification using segment level decision fusion. In *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE, 1286–1290.

[38] Muhammad Bilal Shaikh, Douglas Chai, Syed Mohammed Shamsul Islam, and Naveed Akhtar. 2024. Multimodal fusion for audio-image and video action recognition. *Neural Computing and Applications* 36, 10 (2024), 5499–5513.

[39] Shriman Narayan Tiwari, Ngoc QK Duong, Frédéric Lefebvre, Claire-Helène Demarty, Benoit Huet, and Louis Chevallier. 2016. Deep features for multimodal emotion classification. (2016).

[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[41] Juan Vazquez-Rodriguez, Grégoire Lefebvre, Julien Cumin, and James L Crowley. 2022. Transformer-based self-supervised learning for emotion recognition. In *2022 26th International Conference on Pattern Recognition (ICPR)*. IEEE, 2605–2612.

[42] Johannes Wagner, Andreas Triantafyllopoulos, Hagen Wierstorf, Maximilian Schmitt, Felix Burkhardt, Florian Eyben, and Björn W Schuller. 2023. Dawn of the transformer era in speech emotion recognition: closing the valence gap. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023).

[43] Junwen Xiong, Peng Zhang, Tao You, Chuanyue Li, Wei Huang, and Yufei Zha. 2024. DiffSal: Joint Audio and Video Learning for Diffusion Saliency Prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 27273–27283.

[44] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. 2020. A comprehensive survey on transfer learning. *Proc. IEEE* 109, 1 (2020), 43–76.