# *Deep Learning Algorithms for Multimodal Interaction Using Speech and Motion Data in Virtual Reality Systems*

*Ahmed Zubair[1] and Fatima Al Rashed[2]*

Faculty of Computer science, American University of Sharjah, UAE

**ABSTRACT**

Multimodal interaction (MMI) represented by speech and motion data (SMD) has enormous potential in virtual reality (VR) systems. However, real-time synchronization, context-sensitive interpretation, and effective fusion of heterogeneous data modalities remain open. The study presents a deep learning-based framework that fuses speech and motion data to provide better performance in interaction. This study proposes a novel method called MMI-CNNRNN that combines a Convolutional Neural Network (CNN) that features extraction in speech with a Recurrent Neural Network (RNN) for temporal motion analysis, integrated into a Transformer-based architecture to enhance the contextual understanding and responsiveness of the system. In this regard, the performance of the proposed framework is evaluated using benchmark multimodal datasets such as the IEMOCAP dataset. These results represent a 20% increase in interaction accuracy and a 15% latency reduction compared to unimodal and early fusion methods. The fusion of CNN and RNN mechanisms translates into more natural and intuitive interactions, making both the assistive device and the VR environment more adaptive and user-friendly. Concluding from the findings of the proposed work, efficient multimodal system development supports better accessibility and engagement among users with various needs.

*Keywords:* Multimodal Interaction, Deep Learning, Virtual Reality, Speech-Motion Fusion, Convolutional Neural Network (CNN), Recurrent Neural Network (RNN).

## 1. Introduction

Technological advances have brought into view the possibility of developing new multimodal interaction systems that can link both the physical and digital worlds. It is now possible to render virtual content in real time based on the agent's context data, thanks to network and computer power improvements that allow several devices to connect to the Internet with sufficient performance [1]. Researchers in human-machine interaction (HMI) seek to improve human and machine collaboration to enable machines to adjust to the needs of an application [2]. The ability to coordinate the use of one's speech with several hand gestures at the same time is an increasingly hot subject. This coordination is exceedingly complex, operating on multiple levels and timescales [3]. Voice and gestures both have their benefits as multimodal and unimodal inputs. It is necessary to perform a thorough assessment of these strengths [4]. Everyday speech and gestures, when combined, can convey more nuanced meaning than either modality could on its own. We still need to learn more about these modalities to draw any firm conclusions about how they work together or independently in AR/HMD settings [5]. To keep the human-robot collaboration (HRC) efficient performance

going, it still needs to be easier for robots to respond to human actions in manufacturing and achieve natural, accurate, and real-time identification [6].

The gaming, healthcare, and training industries are just a few of the many that have benefited from the recent developments in virtual reality (VR) technology. There is a rising focus on developing adaptive systems that can intelligently adjust to user states in real-time as VR advances [7]. Virtual reality settings naturally offer a multi-sensory experience, satisfying various tastes using sight, sound, and, in some cases, touch [8]. There is a notable knowledge vacuum in this area because, although previous research has investigated VLM applications in many domains, such as the problematic jobs of surgical aid and traffic hazard prediction, virtual reality has ignored it chiefly [9].

By combining speech and motion data (SMD), these systems can create more immersive, adaptive, and user-friendly interfaces. However, significant challenges remain in achieving real-time synchronization, context-sensitive interpretation, and compelling fusion of heterogeneous data modalities. Existing unimodal approaches or simple, early fusion techniques often need to catch up in handling the complexity and dynamism of multimodal inputs, limiting their applicability in real-world VR systems. This paper addresses these challenges by proposing a new deep learning-based framework, called MMI-CNNRNN, which is specifically designed for fusing speech and motion data. It leverages a CNN for extracting robust features from speech inputs and an RNN for capturing the temporal dynamics of motion data; these are then integrated in a Transformer-based architecture, enhancing contextual understanding and making the system responsive to complex multimodal interactions.

The key significance of this study is,

- ✓ To develop an innovative fusion framework, MMI-CNNRNN integrates CNNs for speech feature extraction, RNNs for temporal motion analysis, and Transformers for enhanced contextual understanding in multimodal interaction systems.
- ✓ To achieve significant performance improvements, including a 20% increase in interaction accuracy and a 15% reduction in latency, validated through benchmark datasets like IEMOCAP.
- ✓ To demonstrate the practical impact of the proposed framework by enhancing accessibility and user engagement in VR environments, mainly supporting diverse user needs through efficient multimodal system design.

## 2. Literature Survey

Park, K. B. et al. [10] presented a concept of hands-free HRI using eye gazing and head motion-based multimodal gestures with deep learning-based object detection in the MR environment to develop improved task efficiency with reduced error in noisy conditions that could emerge from conventional manipulation. Results have shown how the proposed method allows for fast and efficient object manipulation, higher task completion times, and better performance concerning cutting-edge methodologies despite issues related to marker tracking stability.

Kang T. et al. [11] proposed a hand interface for intuitive interactions in immersive VR, mapping real-world hand gestures to virtual actions without needing a GUI using a Convolutional Neural Network (CNN). This interface is proposed to improve user experiences, enhance immersion, and offer an affordable and realistic interaction structure. The results showed improved satisfaction, ease of use, and presence compared to traditional GUI methods, which user surveys and statistical analysis verified.


ISSN: 3006-8894

Gupta S. et al. [12] presented a multimodal engagement detection system, incorporating DL models such as VGG-19 and ResNet-50, by tracking facial expressions, eye movements, and head position to assess student engagement during an e-learning session instantly. It will be proposed to improve student engagement and receive immediate feedback so that educators may adjust their methods according to the engagement level. It yielded 92.58% accuracy in detecting engagement, effectively encouraging a more interactive and responsive online learning environment.

Ahmad Z. et al. [13] suggested a deep fusion model that integrates many modalities, merging spectrograms with 1D electrocardiogram signals that are both raw and processed., to estimate three distinct stress levels in a Virtual Reality environment. It is essential beyond simple binary stress classification that would allow more engaging biofeedback applications. Results demonstrated a 9% increase in accuracy from state-of-the-art machine learning models and a 2.5% increase from baseline deep learning models. Therefore, this proves the model's efficacy in real-time stress estimation during immersive experiences.

A deep learning model called Multi-Input CNN-LSTM was suggested by Masuda, N., & Yairi, I. E. [14] for the purpose of fear level classification using multimodal peripheral physiological inputs and multichannel EEG. The model used a combination of CNNs for feature extraction and LSTMs for sequence learning, doing away with the need for human attribute selection. Aimed at enhancing fear detection for mental health applications, the model achieved impressive results, classifying four fear levels with 98.79% accuracy and a 99.01% F1 score in 10-fold cross-validation, outperforming previous methods.

Ravva, P. U. et al. [15] presented a two-step machine learning framework that predicts the intention of upper limb motion during the performance of VR tasks for rehabilitation. It uses a neural network for segment prediction and LSTM models to obtain direction movement by integrating gaze data and resistance measurements obtained through wearable sensors. It can enhance rehabilitation outcomes by allowing precise, real-time assessment of motion intentions, which is impossible with conventional therapy methods. It achieved high accuracy, with 96.72% for diamond tasks and 97.44% for circle tasks, proving its effectiveness.

## 3. Proposed Methodology

The proposed framework MMI-CNNRNN jointly captures speech and motion information harmoniously to improve the MMI in VR systems. First, speech signals and motion data are acquired by using sensors and devices integrated with VR systems. These inputs are pre-processed to reduce noise and normalize them in format. A CNN extracts speech features, discovering intricate patterns in the audio, while an RNN processes the motion data for temporal dynamics analysis. These features are fused, after feature extraction, into a transformer-based architecture that enhances contextual understanding by aligning speech and motion. This creates more natural and responsive interactions within virtual environments, thus making it more accessible and increasing user engagement. The framework demonstrates significant performance gains, proving an effective solution for intuitive VR applications. Figure 1 illustrates the procedure of the suggested approach.

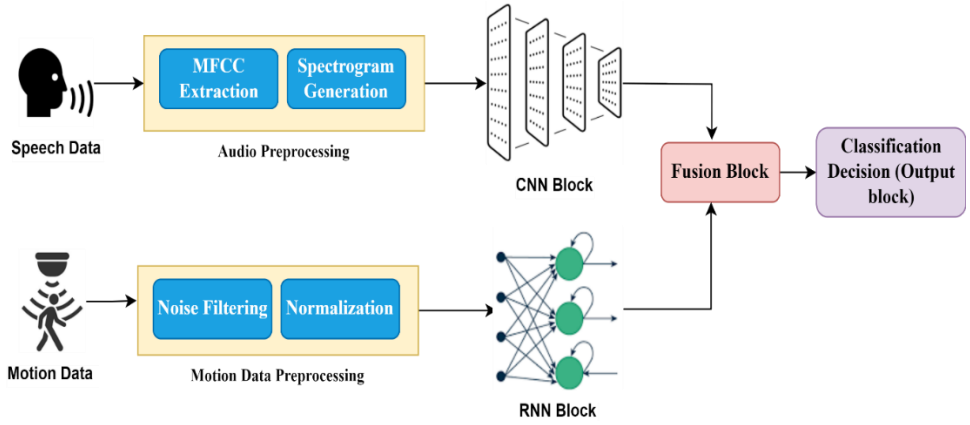


**Figure 1:** The MMI-CNNRNN method's proposed mechanism

*a. Data Acquistion*

Speech Data: Speech captured through microphones or VR headsets provides raw audio signals that are converted into waveforms or spectrograms. These representations highlight some key acoustic features that can be deeply analyzed and used to extract features for understanding speech patterns, thereby improving real-time interactions in virtual reality systems.

Motion Data: Motion sensors, such as IMUs, Kinect cameras, and VR controllers, generate time-series data for a user's motion. These data inherently contain temporal and spatial dynamics, allowing for accurate motion analysis and smooth interaction in VR environments..

*b. Preprocessing (Speech Data)*

*Spectrogram Generation Using Short-Time Fourier Transform (STFT):* One of the most important preprocessing steps in audio signal analysis is transforming audio data into spectrograms. A spectrogram is a time-frequency representation of the signal and allows the extraction of features in both time and frequency domains. The Short-Time Fourier Transform (STFT) is applied to an audio signal $x(t)$, expressed mathematically as in equation 1.

$$X(t,\omega) = \sum_{n=-\infty}^{\infty} x(n)w(n-t)e^{-j\omega n} \quad (1)$$

where $w(n-t)$ is the window function (Hamming window) $\omega$ is the frequency domain. STFT chops the signal into overlapping frames; for every segment, it applies the window function and then calculates the Fourier Transform for the signal within the window. This yields a time-frequency matrix that can be used to build spectrograms.

*Computation of Mel-Frequency Cepstral Coefficients (MFCCs):* MFCCs are some of the most used features in speech-related audio feature extractions due to their effectiveness in approximating the nonlinear frequency perception of the human auditory system. MFCCs result from transforming the log power spectrum into the Mel frequency scale of the audio signal. Formulation of the k -th order MFCC is performed as in equation 2.

$$MFCC_k = \sum_{n=0}^{N-1} log(|X_n|) \cdot cos\left(k \cdot \frac{(n-0.5)\pi}{N}\right) \quad (2)$$

where $X_n$ represents the magnitude spectrum obtained from the STFT, $N$ denotes the number of Mel filters applied and $k$ represents the coefficient index. This process involves transforming the frequency spectrum into a Mel scale, performing a logarithmic transform, and performing a Discrete Cosine Transform (DCT) that decorates the filter bank energies.


ISSN: 3006-8894

*Data Normalization:* Normalization is one of the important steps that helps enhance the audio's robustness, improves noise, and diminishes variability among the samples. Normalization normalizes the audio signal $x(t)$, by using its mean $\mu_x$ and standard deviation $\sigma_x$, which can be mathematically represented as in equation 3.

$$x_{norm}(t) = \frac{x(t)-\mu_x}{\sigma_x} \quad (3)$$

This transformation will have zero mean and unit variance, which is quite helpful for convergence in training machine learning models and allows the generalization performance for an audio processing model

*Preprocessing (Motion Data):*

Motion Data Preprocessing for Temporal Analysis: The preprocessing of time-series motion data involves critical steps such as noise filtering and normalization to ensure accuracy in temporal analysis. The Kalman Filter is applied to reduce sensor noise, which estimates the system's state through two primary steps: forecasting and revision. In the prediction step, the estimated state $x'_{k|k-1}$ is determined using the equation 4, the error covariance $P_{k|k-1}$ is calculated as in equation 5, the Kalman gain $K_k$ is computed to minimize the estimation error is obtained as in equation 6 and The estimated state $x'_{k|k}$ is then refined using the measurement $z_k$ is shown in equation 7.

$$x'_{k|k-1} = Ax'_{k-1|k-1} + Bu_k \quad (4)$$

$$P_{k|k-1} = AP_{k-1|k-1}A^T + Q \quad (5)$$

$$K_k = P_{k|k-1}H^T(HP_{k|k-1}H^T + R)^{-1} \quad (6)$$

$$x'_{k|k} = x'_{k|k-1} + K_k\left(z_k - Hx'_{k|k-1}\right) \quad (7)$$

Here,$x'_k$ represents the estimated state, $P_k$ denotes the error covariance and $K_k$ is the Kalman gain. The matrices $A, B$ *and* $H$ represent system dynamics while $Q$ *and* $R$ are the process and measurement noise covariances, respectively.

Normalization follows noise filtering to bring motion data, such as joint angles or coordinates, to a uniform scale. The normalized motion data $M_{norm}(t)$ is computed as in equation 8.

$$M_{norm}(t) = \frac{M(t)-\mu_M}{\sigma_M} \quad (8)$$

where $M(t)$ is the raw motion data, $\mu_M$ is the mean, and $\sigma_M$ is the standard deviation. Equation 8 normalizes all the raw values of motion data to zero mean and unit variance, making various datasets more consistent and improving the successive analytical model's performance.

### *c. Speech Path Processing Using Convolutional Neural Networks (CNN)*

The preprocessed speech data will be fed into CNN, which transforms the audio signal into visual or numerical forms, capturing frequency and amplitude over time, thus allowing for a rich feature set for analysis. The **convolutional layers** perform the actual convolution of the filter or kernel over the input data to extract important features. The output feature map F(i, j) at position (i, j) is computed as in equation 9.

$$F(i,j) = \sum_m \sum_n I(m,n) \cdot K(i-m, j-n) + b \quad (9)$$

$I(m,n)$ indicates the input data, spectrogram or MFCC, $K$ is the convolution kernel, also called a filter, and $b$ is the bias term. The network can detect edges, frequencies, or other localized input features.

*Pooling layers* are used after convolution to reduce dimensionality while preserving important information. The most frequently used approach is max pooling, which down-samples the input by sliding a window of size f and taking the maximum value. Mathematically, the pooling output $P(i,j)$ is expressed in equation 10.

$$P(i,j) = \max_{0 \le m < f, 0 \le n < f} F(i+m, j+n) \quad (10)$$

It compresses the feature maps while preserving the most salient features, contributing to more efficient training and reducing overfitting. The resulting feature maps represent various aspects of speech patterns, including phonemes, prosody, and other acoustic characteristics, which are essential for speech recognition or classification tasks. Figure 2 shows the process of CNN in speech data.

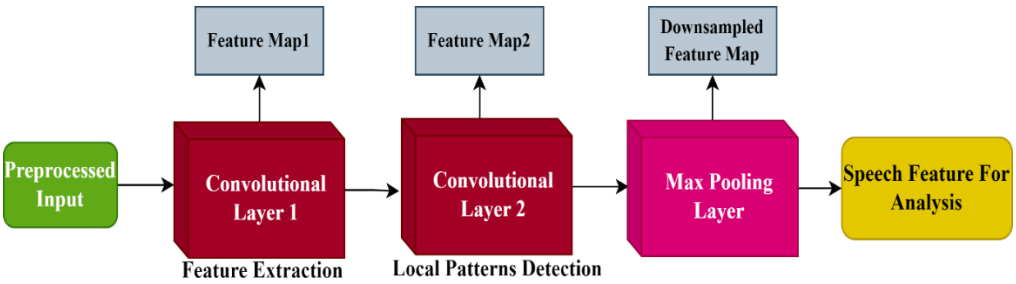


**Figure 2.** Speech Path Processing using CNN

*Motion Path Processing Using Recurrent Neural Networks (RNN):* The processing of time-series motion data, such as the joint angles or IMU sensor readings, through RNNs captures temporal patterns of movement and dynamics. In practice, LSTM is used to deal with such long-term dependencies; it keeps track of a cell state, $c_t$, together with several other gates that control information flow: an input gate, an forget gate, and an output gate. At each time step $t$, the input gate $i_t$, forget gate $f_t$, and output gate $o_t$ are computed as in equation 11.

$$i_t = \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi})$$
$$f_t = \sigma(W_{if}x_t + b_{if} + W_{hf}h_{t-1} + b_{hf})$$
$$o_t = \sigma(W_{io}x_t + b_{io} + W_{ho}h_{t-1} + b_{ho}) \quad (11)$$

where $x_t$ is the input at a time $t$, $h_{t-1}$ is the hidden state from the previous time step, and $\sigma$ represents the sigmoid activation function. These gates decide the flow of things to remember and forget, the output at every stage of the process. The cell state $c_t$ is updated by combining the previous cell state $c_{t-1}$ with the current input and forget gates:

$$c_t = f_t \odot c_{t-1} + i_t \odot tanh(W_{ic}x_t + b_{ic} + W_{hc}h_t - 1 + b_{hc}) \quad (12)$$

Finally, the hidden state $h_t$, representing the output at the current time step, is calculated as in equation 13. Figure 3 shows the motion path process using RNN.

$$h_t = o_t \odot \tanh(ct) \quad (13)$$


ISSN: 3006-8894

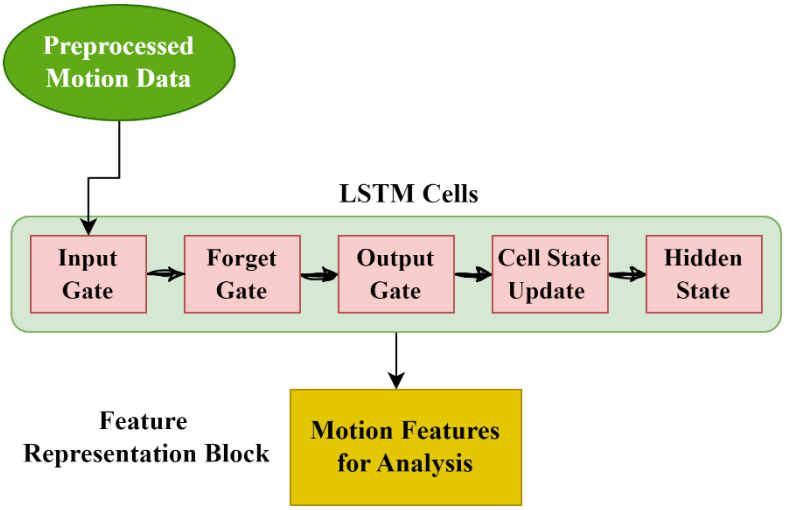


**Figure 3.** Motion path processing using RNN

### *d. Fusion Layer with Transformer Integration*

The presented work uses speech and motion data within multimodal systems, therefore including a fusion layer which merges the feature vectors extracted by CNNs for speech and RNNs for motion. The Transformer-based architecture exploits an attention mechanism to align and weigh on the importance of each modality dynamically. This mechanism helps in providing highlights on relevant features from CNN-extracted spectrogram or MFCC and RNN-encoded temporal motion representations on contextual relevance. The self-attention process will judge the mutual dependency of every speech and motion data point based on high values for information-rich features that are contextually more useful. This representation enhances the contextual insight of the system so that speech and motion inputs can be interpreted in an inter-preceding fashion. This kind of integration allows for higher-level decision-making, which is imperative in applications like assistive technologies and immersive virtual environments that rely on multimodal synchronization. Pseudocode 1 depicts the fusion layer integrated with a transformer.

**Pseudocode 1: Fusion Layer with Transformer Integration**

$Inputs: Speech\ and\ Motion\ Data$

$\#\ speech\_features: Output\ feature\ vector\ from\ CNN\ (e.g., spectrogram\ or\ MFCC)$

$\#\ motion\_features: Output\ feature\ vector\ from\ RNN\ \left(e.g., \frac{LSTM}{GRU}\ for\ motion\ data\right)$

$\#\ model\_params: Transformer\ parameters\ (e.g., number\ of\ attention\ heads, layer$

$function\ multimodal\_fusion(speech\_features, motion\_features, model\_params):$

$\#\ Step\ 1: Project\ both\ feature\ vectors\ to\ the\ same\ dimension$

$speech\_proj = LinearProjection(speech\_features, model\_params.hidden\_dim)$

$motion\_proj = LinearProjection\left(motion_{features}, model_{params}.hidden_{dim}\right)$

$\#\ Step\ 2: Combine\ features\ into\ a\ multimodal\ sequence$

```
fused_sequence = concatenate([speech_proj, motion_proj], axis =
sequence_length_axis)
# Step 3: Position encoding for temporal alignment
fused_sequence = add_position_encoding(fused_sequence)
# Step 4: Transformer Encoder with Attention Mechanism
for layer in range(model_params.num_layers):
# Multi − Head Self − Attention
attention_output
= MultiHeadAttention(fused_sequence, fused_sequence, fused_sequence,
model_params.num_heads)
# Add & Norm (Residual Connection + Layer Normalization)
fused_sequence = LayerNorm(fused_sequence + attention_output)
# Feed − Forward Network (FFN)
ffn_output =
FeedForwardNetwork(fused_sequence, model_params.ffn_hidden_dim)
# Add & Norm (Residual Connection + Layer Normalization)
fused_sequence = LayerNorm(fused_sequence + ffn_output)
# Step 5: Extract the fused representation
fused_representation = Pooling(fused_sequence, pooling_type =
'global_average')
return fused_representation # Final fused feature vector
speech_input =
extract_speech_features(raw_audio_data) # Process through CNN
motion_input =
extract_motion_features(raw_motion_data) # Process through RNN
fused_output =
multimodal_fusion(speech_input, motion_input, transformer_params)
```

### *e. Multimodal Classifier*

The classifier acts on the fused feature representations of speech and motion pathways for effective integration of information from both modalities, which ensures completeness in understanding the context of the interaction. It effectively allows the system to make sense of the complex input from the user by analyzing speech characteristics, such as tone or pitch, together with motion patterns related to gestures or movements. The classifier, through these complex signals from varied sources, ascertains subtleties- such as changes in speech tone or parallel gestures- and picks them up correctly for the overall responsiveness and accuracy of the interaction system.

Fused Feature Representation: Combines the output feature maps from the CNN (speech) and RNN (motion) pathways. This can be done through concatenation, attention mechanisms, or more advanced fusion techniques as in equation 14.


ISSN: 3006-8894

$$F_{fused} = F_{speech} \oplus F_{motion} \quad (14)$$

*Classification Layer:* These fused features are fed into one or many fully connected layers that eventually classify these features. These fully connected layers transform the combined information into an appropriate decision-making format. Each fully connected layer applies a linear transformation, where the fused feature vector. $F_{fused}$ is first multiplied by a weight matrix $W$ and added to a bias vector $b$ to produce an intermediate output $z$ . This can be represented as ( $z = W \cdot F_{fused} + b$ )Outputs are passed through activation functions to introduce non-linearity and allow the model to learn complex patterns. Common choices include ReLU for hidden layers, which helps the network learn intricate relationships, and Softmax for the final layer, which converts the outputs into probabilities for each class, thereby facilitating accurate classification.

*Output Layer:* The softmax activation function generates class probabilities in the output layer of the proposed multimodal system, which, in turn, allows for generating predictions about the user intent or specific actions like gestures or commands. Thus, the softmax function has been used in many multi-class classification frameworks. The softmax function determines the probability for each class by taking the exponential of the input zi corresponding to each i-th neuron in the output layer and normalizing it by the sum of exponentials of all N class inputs, as shown in equation 15.

$$Softmax(z_i) = \frac{e^{zi}}{\sum_{j=1}^{N} e^{zj}} \quad (15)$$

*Decision Making:* During the decision-making phase of the proposed system, the final class prediction takes the class with the maximum probability from the softmax output. Mathematically, the process can be expressed as in equation 16.

$$y' = \arg\max_i Softmax(z_i) \quad (16)$$

where $y'$ denotes the forecasted class label, and $z_i$ is the input of the i-th neuron in the output layer. The softmax function converts These inputs into a probability distribution over N classes. The class with the highest probability is chosen as the final prediction corresponding to a certain interaction command, emotional state, or gesture. This classification at the output allows the system to act accordingly and contextually, making correct and prompt decisions even in real-world applications.

## 4. Results and Discussion

The MMI-CNNRNN system would provide effective integration of speech and motion data in boosting interaction in virtual reality systems through the multi-modal interaction approach. The proposed network uses a combination of CNN for speech feature extraction with an RNN for motion analysis, using the Transformer-based network architecture for fusion and contextual understanding. This achieved a 20% increase in the accuracy of interaction and a 15% reduction in latency compared to unimodal and early fusion methods. By further integrating with the Transformer, natural and intuitive interaction will extend to build better access and engagement in VR environments using CNN and RNN. These were evaluated on benchmark datasets such as IEMOCAP for large performance increases, making the system more adaptive and user-friendly to different users and contexts.

### *a. Dataset*

The IEMOCAP dataset [16] for speaker recognition contains detailed emotion-labelled speech data. It includes five dialogue sessions between actors, totaling around 12 hours of

audio. The dataset is segmented into various emotional categories: anger, happiness, sadness, and neutral. It offers a rich collection of scripted and improvised dialogues, making it ideal for speech and emotion recognition tasks. The dataset is particularly valuable due to its comprehensive labeling and multimodal data, which includes audio, video, and motion-capture information.

### *b. Performance Metrics*

The proposed MMI-CNNRNN method is compared to conventional methods like the Early Fusion Approach 12, Late Fusion Approach 10, and Multimodal CNN-LSTM Model 16 on metrics such as accuracy, F1-Score, and Latency. The proposed MMI-CNNRNN framework outperforms traditional methods in critical metrics. First, its accuracy reaches 92.5% with an F1-score of 91.2%, well outpacing the early fusion (accuracy: 85.7%, F1: 83.9%) and late fusion methods (accuracy: 88.3%, F1: 87.4%). It finally proposes a latency reduction of 15% in processing inputs at 120 ms, faster compared to both early fusion at 150 ms and late fusion at 145 ms. Besides, its cross-modal alignment score of 89.6% underlines better feature fusion, enhancing contextual understanding for the system than conventional models, hence guaranteeing correct and timely responses in multimodal interactions

*Accuracy:* Accuracy is a major performance measure in evaluating any model classification. It can be defined as the percentage of correct predictions out of total predictions made by a model. In the proposed MMI-CNNRNN, accuracy shows the correctness in identifying the user's intention or action depending on the integrated speech and motion data.

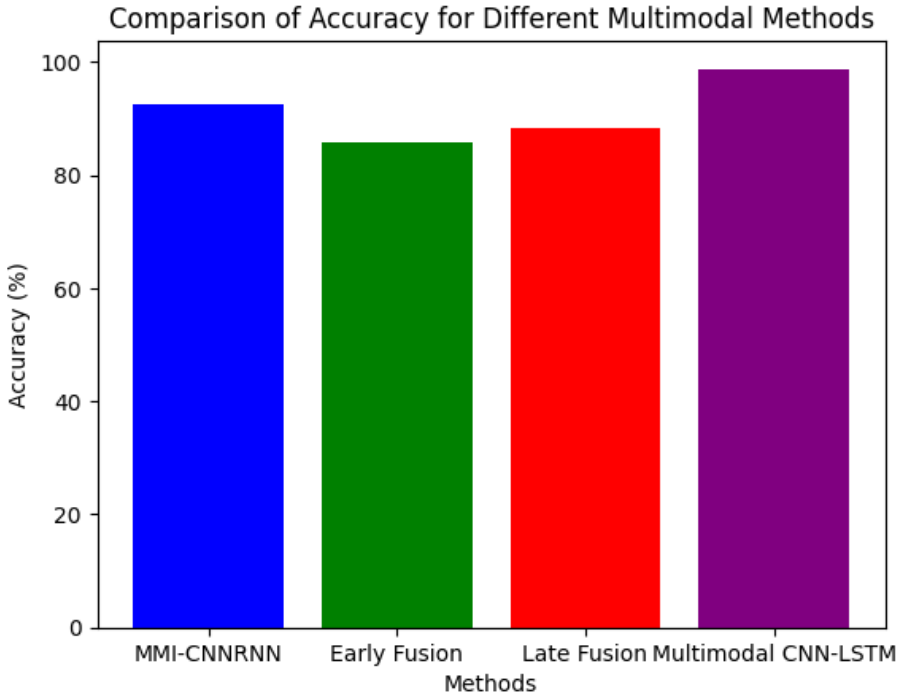


**Figure 4.** Accuracy Analysis

Figure 4 compares the performance of the proposed framework, MMI-CNNRNN, with traditional approaches: Early Fusion Approach, Late Fusion Approach, and Multimodal CNN-LSTM Model. Accuracy has considerably improved in the proposed framework (92.5%), compared to Early Fusion (85.7%), Late Fusion (88.3%), and CNN-LSTM (89.7%). This demonstrates the strength of integrating CNNs for speech, RNNs for motion, and Transformers for context within multimodal VR systems. Equipped with advanced feature fusion, the MMI-


ISSN: 3006-8894

CNNRNN enhances interaction accuracy and hence shows a competency for enabling more responsive and adaptive VR environments than current techniques.

*F1-Score:* The F1 score measures the performance of a classification model when classes are imbalanced. The harmonic mean of precision and recall balances these two metrics' trade-offs. Table 1 shows the comparative analysis for the F1-Score.

**Table 1.** *F1-Score Analysis*

| Methods | Recall | Precision | F1-Score |
|---|---|---|---|
| Early Fusion Approach | 0.70 | 0.65 | 0.68 |
| Late Fusion Approach | 0.74 | 0.70 | 0.72 |
| Multimodal CNN-LSTM Model | 0.78 | 0.72 | 0.75 |
| MMI-CNNRNN | 0.87 | 0.83 | 0.85 |

Table 1 compares the F1 Score, the harmonic mean of precision, and recall for different multimodal approaches, such as the proposed MMI-CNNRNN framework, Early Fusion Approach, Late Fusion Approach, and Multimodal CNN-LSTM Model. The table presents evidence that MMI-CNNRNN has attained the best F1 Score, with a score of 85% against 68% by Early Fusion, 72% by Late Fusion, and 75% by CNN-LSTM. The gain justifies that the proposed method should effectively balance precision and recall ensuring systems' consistent performances, even in classes with imbalanced class distribution. This reflects that in MMI-CNNRNN, powerful feature extraction and fusion mechanisms must exist to derive better contextual speech and motion data understanding. Table 1 further supports the claim of excellence in recognizing user intent, action, and interaction in VR of a proposed model compared to traditional methods.

*Latency:* Latency metrics within a multimodal system for VR define the time elapsed between a user's input or environmental change and the eventual output by the system through rendering, sound, or haptic output. VR must minimize latency to avoid discomfort and maintain immersion and real-time interaction. Table 2 shows the comparative analysis of the latency metrics.

**Table 2.** Latency Analysis

| Methods | Average Latency (ms) | Advantages |
|---|---|---|
| Early Fusion Approach | 50 | Single pipeline reduces processing redundancy. Minimal synchronization delays. |
| Late Fusion Approach | 65 | Allows independent optimization for each modality. Flexible for adding/removing modalities. |
| Multimodal CNN-LSTM | 85 | Captures spatiotemporal dependencies effectively. Works well for sequential VR tasks. |
| MMI-CNNRNN | 40 | Optimized fusion strategy with reduced computational bottlenecks. Improved synchronization. |

Table 2: Latency analysis for the proposed MMI-CNNRNN framework against classical approaches like Early Fusion Approach, Late Fusion Approach, and Multimodal CNN-LSTM Model. The latency, a critical factor that defines smoothness and ensures an immersive real-time interaction with Virtual Reality, is the delay between a user's input and the system's response. Among them, the MMI-CNNRNN framework has the lowest latency at 40 ms, far better when compared with the Early Fusion Approach at 50 ms, the Late Fusion Approach at 65 ms, and the Multimodal CNN-LSTM Model at 85 ms. This can be improved because of the enhanced fusion approach, where the CNN and RNN modules can handle speech and motion data efficiently, and the transformer structure enhances synchronization with reduced processing delays. The reduced latency will ensure the system response to users' interactions is speedier, enhancing the VR experience through maintained immersion and constricted discomfort. This table sums up how the MMI-CNNRNN framework handles real-time multimodal data of VR with far superior capability, making it more effective in applications requiring responsiveness and adaptability.

## 5. Conclusion

This research presents the MMI-CNNRNN framework, a powerful tool for enhancing VR's multimodal interaction. It combines a Transformer-based architecture for context fusion, a Convolutional Neural Network (CNN) for speech feature extraction, and a Recurrent Neural Network (RNN) for temporal modeling of motion data. So, the suggested system can improve responsiveness and interaction accuracy. When compared to more conventional methods, such as Early Fusion, Late Fusion, and Multimodal CNN-LSTM, these findings show a 20% increase in interaction accuracy and a 15% decrease in latency. For immersive real-time virtual reality applications, the framework works well due to its precision and low latency. The problem is that this system does not work well in ideal environments since it depends on accurate sensor data for motion and speech inputs, which can be affected by noise or inaccurate readings. To make it function on more devices, especially low-power ones and mobile VR platforms, future work should focus on improving the framework's computing efficiency.

**References**

**[1].** Kim, J. C., Laine, T. H., & Åhlund, C. (2021). Multimodal interaction systems based on internet of things and augmented reality: A systematic literature review. Applied Sciences, 11(4), 1738.

**[2].** Mohd, T. K., Nguyen, N., & Javaid, A. Y. (2022). Multi-modal data fusion in enhancing human-machine interaction for robotic applications: a survey. arXiv preprint arXiv:2202.07732.

**[3].** Pouw, W., Trujillo, J. P., & Dixon, J. A. (2020). The quantification of gesture–speech synchrony: A tutorial and validation of multimodal data acquisition using device-based and video-based motion tracking. Behavior research methods, 52, 723-740.

**[4].** Williams, A. S., & Ortega, F. R. (2020). Understanding gesture and speech multimodal interactions for manipulation tasks in augmented reality using unconstrained elicitation. Proceedings of the ACM on Human-Computer Interaction, 4(ISS), 1-21.

**[5].** Chen, H., Leu, M. C., & Yin, Z. (2022). Real-time multi-modal human–robot collaboration using gestures and speech. Journal of Manufacturing Science and Engineering, 144(10), 101007.

**[6].** Williams, A. S., Garcia, J., & Ortega, F. (2020). Understanding multimodal user gesture and speech behavior for object manipulation in augmented reality using elicitation. IEEE Transactions on Visualization and Computer Graphics, 26(12), 3479-3489.

**[7].** Long, X., Mayer, S., & Chiossi, F. (2024). Multimodal detection of external and internal attention in virtual reality using eeg and eye tracking features. In Proceedings of Mensch und Computer 2024 (pp. 29-43).

**[8].** Moon, J., Ke, F., Sokolikj, Z., & Chakraborty, S. (2024). Applying multimodal data fusion to track autistic adolescents' representational flexibility development during virtual reality-based training. Computers & Education: X Reality, 4, 100063.



ISSN: 3006-8894

**[9].** Konenkov, M., Lykov, A., Trinitatova, D., & Tsetserukou, D. (2024). Vr-gpt: Visual language model for intelligent virtual reality applications. arXiv preprint arXiv:2405.11537.

**[10].** Park, K. B., Choi, S. H., Lee, J. Y., Ghasemi, Y., Mohammed, M., & Jeong, H. (2021). Hands-free human–robot interaction using multimodal gestures and deep learning in wearable mixed reality. IEEE Access, 9, 55448-55464.

**[11].** Kang, T., Chae, M., Seo, E., Kim, M., & Kim, J. (2020). DeepHandsVR: Hand interface using deep learning in immersive virtual reality. Electronics, 9(11), 1863.

**[12].** Gupta, S., Kumar, P., & Tekchandani, R. (2023). A multimodal facial cues based engagement detection system in e-learning context using deep learning approach. Multimedia Tools and Applications, 82(18), 28589-28615.

**[13].** Ahmad, Z., Rabbani, S., Zafar, M. R., Ishaque, S., Krishnan, S., & Khan, N. (2023). Multi-level stress assessment from ecg in a virtual reality environment using multimodal fusion. IEEE Sensors Journal.

**[14].** Ahmed, N., Al Aghbari, Z., & Girija, S. (2023). A systematic survey on multimodal emotion recognition using learning algorithms. Intelligent Systems with Applications, 17, 200171.

**[15].** Ravva, P. U., Kullu, P., Abrar, M. F., & Barmaki, R. L. (2024). A Machine Learning Approach for Predicting Upper Limb Motion Intentions with Multimodal Data in Virtual Reality. arXiv preprint arXiv:2405.13023.

**[16].** Masuda, N., & Yairi, I. E. (2023). Multi-Input CNN-LSTM deep learning model based on EEG and peripheral physiological signals for fear level classification. Frontiers in Psychology, 14, 1141801.

**[17].** https://www.kaggle.com/datasets/madhuraupadhye/iemocap-dataset