Chapter 6

Enhancing Dynamic Hand gesture recognition with sEMG-Based Multi-Modal Sensors Fusion

In the previous chapter, we introduced a machine learning framework for dynamic gesture recognition, with a focus on handwriting gesture recognition using surface electromyography (sEMG). During our experiments, we observed that while sEMG effectively captures muscle activation patterns, it lacks the ability to capture kinematic information. Adding kinematic information and extracting patterns from dynamic gestures seem to us to improve the accuracy of dynamic hand gesture recognition. Recognizing this as an area for improvement, we incorporated an Inertial Measurement Unit (IMU) with sEMG signals to gather the necessary kinematic data from dynamic gestures.

To address this, we implemented a multi-modal sensor fusion approach that combines data from both sEMG and IMU sensors. This integration was achieved using a modified deep autoencoder to create an effective representation of the combined sensory data. For our analysis, we collected a new dataset specifically for handwriting gesture recognition on a whiteboard. This dataset comprises raw signals

corresponding to the 26 lowercase English alphabets, written on a whiteboard by multiple subjects. It was specifically created due to the lack of any publicly available datasets of this kind, to the best of our knowledge. Building on our initial findings on sEMG-based dynamic gesture recognition, this chapter seeks to further enhance the performance of our model by integrating IMU data through a multi-modal feature fusion approach. The details of this approach are covered in the following subsections of this chapter.

Section 6.1 introduces the chapter, presenting the problem statement, outlining the proposed solution, and the research questions addressed. Section 6.2 outlines the foundational aspects of our approach, including a discussion on our novel dataset collection, data preprocessing strategies involving various filters, and the feature extraction techniques employed. Section 6.3 discusses the Deep Autoencoder-based Multi-modal Sensor Feature Fusion, elaborating on the proposed architecture, significant findings, and conclusions. These approaches aim to improve recognition accuracy by leveraging the complementary strengths of sEMG and IMU sensors, providing a more reliable solution for dynamic gesture recognition.

6.1 Introduction

Multi-modal deep architecture has demonstrated significant advantages across various applications by integrating diverse data sources to enhance performance. These complex architectures have been utilized in various applications for improved performance, including medical imaging [87,88], computer vision for tasks like image captioning, visual question answering [89, 90], and speech recognition for phoneme sequence & audio signal [91, 92]. In natural language processing, the multi-modal deep learning models have been instrumental in speech-to-text translation [93] and audio captioning [94]. It has also found applications in robotics and autonomous vehicles [95,96], where they facilitate the fusion of sensor data from diverse modalities

such as visual, lidar, and radar. These fusion processes have been credited with enhancing the accuracy of perception and decision-making within these systems [97].

Inspired by the benefits of multimodal sensing and sensors-based approaches, we used Electromyography (EMG) and Inertial Measurement Unit (IMU) sensors to gather data corresponding to different handwritten English characters. These sensors complement each other to track the various aspects of handwriting recognition tasks. For instance, EMG tracks the muscle's activation pattern produced while writing characters [84], while IMU can track the motion with acceleration, gyroscope, and magnetometer [109].

We conducted feature-level fusion to leverage the capabilities of both sensing modules. This fusion aims to enhance the accuracy of handwriting recognition by an improved shared representation using the proposed deep autoencoder variant.

6.1.1 Problem statement and overview of the solution

6.1.1.1 Problem statement

Given a dataset D, how can we develop an efficient Handwriting Character Recognition (HCR) system that accurately interprets and recognizes handwriting using data fusion from surface electromyography (sEMG) and inertial measurement units (IMU), ensuring feasibility for enhancing digitization and related tasks in various applications?

Where, Dataset D is represented as $D = \{d_j \mid j = 1, \ldots, N\}$, where N is the number of data instances in D. Each data instance $d_j = (I_j, S_j, L_j)$ is a tuple consisting of a time-series I_j representing the IMU sensor data, a time-series S_j representing the EMG sensor data, and a class label $L_j \in \{0, \ldots, 26\}$.For our models, we aim to use both the IMU and EMG sensor data.

IMU Sensor Data: Every $I_j = \{I_j^{(1)}, I_j^{(2)}, \ldots, I_j^{(T_j)}\}$ is a time-series consisting of T_j time-steps (i.e., $|I_j| = T_j$), and each element $I_j^{(t)} = \left[a_j^t, g_j^t, q_j^t\right]$. The vectors $a_j^t \in \mathbb{R}^3$

and $g_j^t \in \mathbb{R}^3$ are the 3 axes of the accelerometer and the gyroscope, respectively. Similarly, $q_j^t \in \mathbb{R}^4$ denotes a quaternion representing the orientation.

EMG Sensor Data: Every $S_j = \{S_j^{(1)}, S_j^{(2)}, \ldots, S_j^{(T_j)}\}$ is a time-series consisting of T_j time-steps (i.e., $|S_j| = T_j$), and each element $S_j^{(t)}$ is a vector representing the EMG signals from multiple channels. Let $e_j^t \in \mathbb{R}^m$ denote the EMG signal vector at time step t, where m is the number of EMG channels. Thus, $S_j^{(t)} = [e_j^t]$ captures the muscle activity at each time step.

By effectively integrating and processing this multimodal sensor data, we aim to enhance the accuracy and feasibility of the HCR system for various real-world applications.

6.1.1.2 Overview of the solution

As a solution, we proposed an efficient Handwriting Character Recognition (HCR) model that interprets and recognizes human handwriting character patterns using deep feature representation learning [36]. Data collected from multiple sensors are processed and given to a modified autoencoder architecture that inputs multi-modal data parallel. For our HCR pipeline, we initially framed a set of predefined handwriting characters $'\mathcal{C}'$ represented as:

$$\mathcal{C} = \{C_i\}_{i=1}^{w} \tag{6.1}$$

Where $'w'$ denotes the set of different characters written on a whiteboard. Data is collected from multiple sensors and treated as sequences capturing different aspects of the handwriting information, which is considered as follows:

$$
\begin{aligned}
S_1 &= \{d_{11}, d_{12}, \ldots, d_{1t}, \ldots, d_{1n}\} \ \text{(Sensor}_1\text{\_readings)} \\
S_2 &= \{d_{21}, d_{22}, \ldots, d_{2t}, \ldots, d_{2n}\} \ \text{(Sensor}_2\text{\_readings)} \\
&\vdots \\
S_j &= \{d_{j1}, d_{j2}, \ldots, d_{jt}, \ldots, d_{jn}\} \ \text{(Sensor}_j\text{\_readings)}
\end{aligned}
$$

$S_j; 1 \leq j \leq k$ are the different sensors , d_{jt} represents the j^{th} Sensor reading at time t and n denotes the length of the sequence.

In our proposed method, data is collected from two types of sensors: sEMG sensors and IMU sensors. The sEMG sensors capture muscle activation signals from the hand muscles, while the IMU sensors measure the motion and orientation of the hand during handwriting. The goal is to build a model $\mathcal{H}$ that can accurately predict the handwriting character label, $\hat{\mathcal{C}}$, based on the information obtained using sEMG and IMU sensors.

$$
\hat{\mathcal{C}} = \left\{\hat{C}_l\right\}_{l=1}^{v} = \mathcal{H}[\varphi(\mathcal{S}_1, \ldots, \mathcal{S}_j)], \quad \text{where } \hat{C}_l \in \mathcal{C} \text{ and } v \leq w \tag{6.2}
$$

The model $\mathcal{H}$ doesn't take the sequences directly; instead, it takes the output of a projection function φ. For our case, φ represents the feature extraction process. We extracted meaningful features from sEMG and IMU signals (details in Section 3.6). The feature extraction process converts the sequences into d-dimensional vectors, $\varphi(\mathcal{S}_1, \ldots, \mathcal{S}_j) \in \mathbb{R}^d$.

The obtained feature set is optimized, and to achieve feature-level fusion [275] and derive a compressed, meaningful representation, a Multimodal Deep Autoencoder (MDAE) is employed. The MDAE processes concurrent inputs from diverse sEMG and IMU feature vectors, generating deep features.

Deep Feature Learning Using Multimodal Deep Autoencoder (MDAE): The Multimodal Deep Autoencoder (MDAE) processes the combined input of sEMG and IMU feature vectors to generate deep features, denoted as Deep_Features. This process is represented by the equation:

$$Deep\_features = \text{MDAE}(\text{sEMG}, \text{IMU}) \tag{6.3}$$

MDAE utilizes an unsupervised learning approach wherein a deep neural network is trained to reconstruct the input data. It functions as a mapping model that transforms the input data from the various sEMG and IMU features into the corresponding deep feature space. The deep features extracted by the autoencoder are then used for the classification task to predict the handwriting character sequence. The approach is employed with the objective of early feature fusion. Early feature fusion involves concatenating features from different modalities (sEMG and IMU) at the initial stage to form a single feature vector, which is then fed for classification. This unified feature representation ensures that all available information is utilized simultaneously, allowing the model to learn potential interactions and dependencies between modalities from the outset.

Let the true handwriting character sequence (ground truth) be represented as:

$$\mathcal{C}' = \{C'_l\}_{l=1}^{v}, \quad \text{where } C'_l \in \mathcal{C} \tag{6.4}$$

The goal of the Handwritten Character Recognition (HCR) framework is to develop a classifier, $\mathcal{H}$, that correlates the selected feature subset or deep features to the handwritten character class label predictions. This process can be represented as:

$$\hat{\mathcal{C}} = \mathcal{H}[Deep\_features]$$

The motive is to minimize the discrepancy between the predicted character labels $\hat{\mathcal{C}}$ and the true labels $\mathcal{C}'$. This is typically achieved by minimizing a loss function, which measures the difference between the predictions and the ground truth $L(\mathcal{H}(\text{Features}), \mathcal{C}')$. The process of feature fusion using a multi-modal deep autoencoder (MDAE) is outlined in Algorithm 13

Algorithm 13 Deep Feature Learning Using MDAE

1: Initialize the MDAE architecture
2: Pretrain the MDAE on the input features
3: Fine-tune the MDAE using the combined sEMG and IMU features
4: Encode the input features to obtain deep features
5: **return** Deep_Features

The proposed method is illustrated schematically in Figure 6.1.To validate the effectiveness of the proposed model, we formulated the following research questions(RQs):

1. RQ12:*How effective is the fusion of sEMG signals with accelerometer and gyroscope data to improve the accuracy and robustness of dynamic hand gesture recognition? (Addressed in Section 6.3.2)*

2. RQ13:*How does the proposed model perform when deployed on IoT-enabled low-cost devices such as the Raspberry Pi 3? (Addressed in Section 6.3.2.3)*

3. RQ14: *How does the proposed multi-modal sensor fusion approach perform compared to existing state-of-the-art methods in terms of accuracy, robustness, and computational efficiency? (Addressed in Section 6.3.3)*

6.1.2 Major contribution of this work

This chapter's primary contribution can be summarized as follows.

a) We proposed a reliable and efficient Multi-modal handwritten character recognition pipeline that uses feature-level information fusion and can accurately predict different characters written on a whiteboard.

b) To find the representative feature set, we employed a feature fusion approach using a deep autoencoder. The approach generates deep features that prove more effective than the original feature set.

c) We collected a new dataset of large samples corresponding to 26 lowercase English alphabets handwritten on the whiteboard.

d) The proposed pipeline achieved the recognition accuracy of 99.01% for twenty-six lowercase English alphabets, even using baseline classifiers.

e) The effectiveness of the pipeline is thoroughly validated using standard performance metrics, and separate analysis is performed to assess its performance when deployed on IoT-enabled low-cost devices (Raspberry Pi 3).

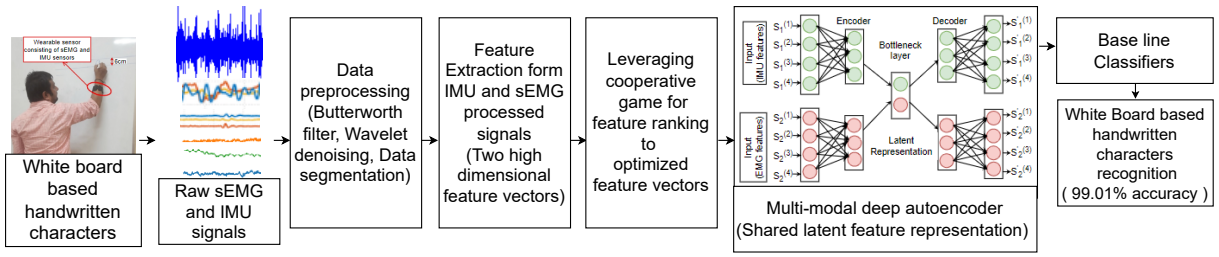

FIGURE 6.1: A schematic diagram illustrating the proposed pipeline for whiteboard-based handwritten character recognition task

6.2 Materials and methods

6.2.1 Classification of Data Fusion techniques

Multi-sensor fusion can be divided into three categories: data-level fusion, feature-level fusion, and decision-level fusion.

Data-level Fusion: This level integrates raw data from multiple sensors to enhance the reliability, robustness, and generalization of the recognition system. When sensors measure the same phenomenon, data can be fused directly. However, for data from heterogeneous sources, advanced fusion techniques are required [275].

Feature-level Fusion: This method combines feature sets extracted from different sensors to form a high-dimensional feature vector used for classification or pattern recognition. It is commonly applied to both heterogeneous and homogeneous sensors [276].

Decision-level Fusion: This involves merging individual decisions from multiple sensors to reach a final decision. It utilizes processed information to achieve high-level decisions and is ideal for systems where different sensors provide complementary information. Techniques such as generalized discriminant analysis (GDA) and multi-class relevance vector machines (RVM) are used to improve the real-time performance and accuracy of recognition systems [277].

The feature-level fusion concept was utilized to design the proposed model, aiming to improve the overall performance of our handwriting recognition system. Feature-level fusion is considered more effective than other fusion levels because the feature set contains more comprehensive information about the input biological characteristics compared to the matching score or the classifier's output decision. Consequently, fusion at the feature level is anticipated to yield better recognition results [278]

6.2.2 Dataset: Multi-Modal Handwritten Character Recognition (MM-HCR) dataset

To evaluate the fusion of sEMG and IMU sensor data for handwritten character recognition, we created the Multi-Modal Handwritten Character Recognition (MM-HCR) dataset. This dataset includes signals from eight sEMG (surface electromyography) sensors and three IMU (inertial measurement unit) sensors—specifically a

gyroscope, accelerometer, and magnetometer. These sensors captured data for the 26 lowercase English alphabet characters written on a whiteboard (details provided in section 2.3.12).

Data Collection Protocol: During data collection, subjects wrote the isolated alphabets on a whiteboard using a marker, and the corresponding sEMG and IMU sensor readings were recorded and stored in CSV format. We adhered to a strict protocol to mitigate muscle fatigue's effects on EMG signals. The data collection was performed over multiple sessions, referred to as recordings. Each recording consisted of various sets, each comprising 10-15 repetitions performed by a subject. Efforts were made to separate each set with a 15-second interval. Following this protocol, we collected around 800 repetitions of each character, resulting in approximately 21,000 samples for the 26 lowercase English alphabets. Figure 6.2 provides a detailed overview of the data collection protocol.

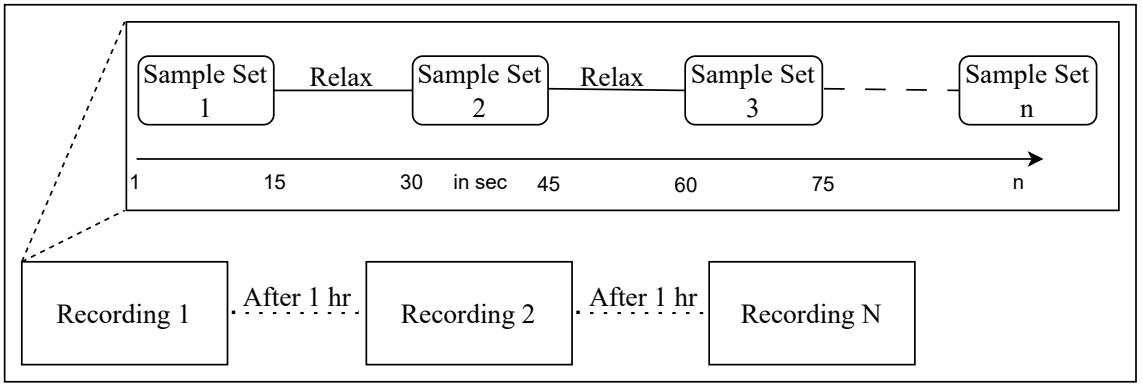

FIGURE 6.2: Data collection protocol

Uniformity and Reference Guidelines: To maintain uniformity of font size, two parallel lines separated by a distance of 6 cm were used as a reference while writing these characters on the whiteboard. A template for all 26 lowercase English alphabets was used as a reference.

Class Labeling Twenty-six classes correspond to the 26 lowercase English alphabets for the recognition task. Each class, denoted as C_k, is assigned a distinct label l_k, where $1 \leq l_k \leq 26$, to facilitate supervised classification.

6.2.3 Signal preprocessing and filters

The raw signals collected for our dataset were pre-processed with digital filters to reduce the effect of noise added during data collection. For the sEMG signals, we applied a Butterworth band-pass filter to extract the dominant frequency. Moreover, a configurable Butterworth filter dealt with 50 Hz power noise [246].

For the IMU, we aimed to reduce noise interference by applying wavelet denoising. This pre-processing step is crucial in improving the accuracy of subsequent analyses and extracting more meaningful information from the IMU data. Figures 6.3 and 6.4 illustrate the raw and filtered signals for sEMG and IMU after applying the mentioned filters.

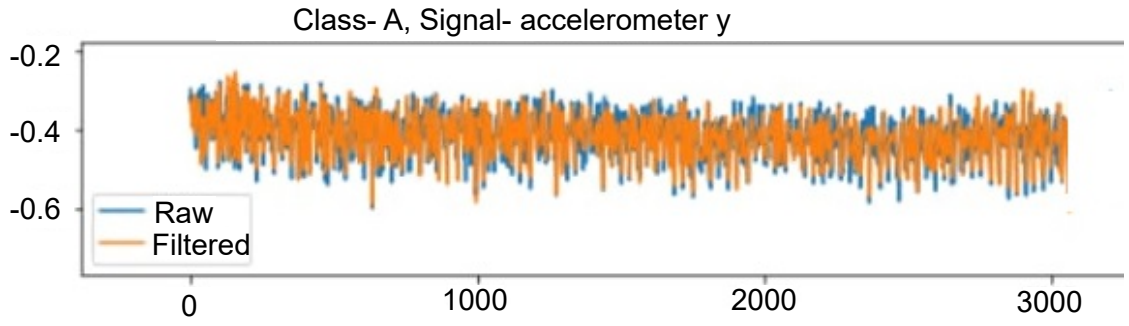

FIGURE 6.3: Raw and filtered accelerometer signal using wavelet denoising

6.2.4 Feature Extraction

The efficacy of myoelectric-based applications relies significantly on the quality of features obtained from raw data. These features are critical in selecting control strategies for prosthetic devices, robotic control systems, and hand gesture recognition. For surface electromyography (sEMG), we have identified key features documented in the literature, particularly those relevant to handwriting recognition and related tasks [217, 279]. Similarly, for inertial measurement unit (IMU) signals, we have adopted a comparable approach to identify pertinent features employed in IMU-based handwriting recognition tasks. The details of the feature are given in section 4.2.4, feature extraction was done using the library [251]

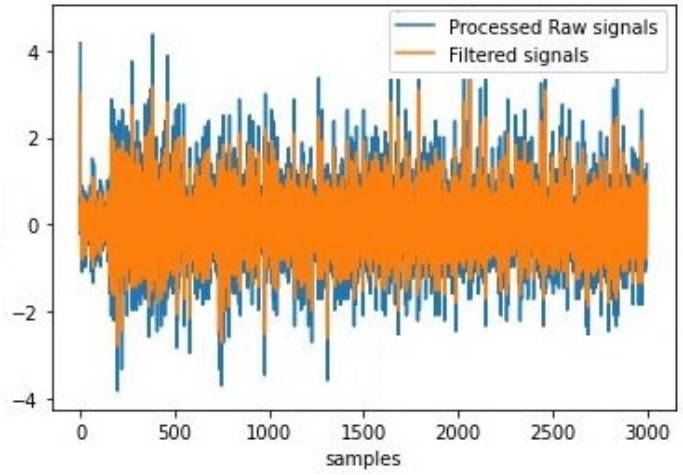

FIGURE 6.4: Raw and filtered sEMG signals using Butterworth filter

6.2.5 Multi-modal Deep Autoencoder for Feature Fusion

Recent advancements in deep learning have shown significant efficacy in feature-level fusion tasks [276]. Autoencoders, in particular, are well-suited for this purpose due to their operational efficiency and architectural properties [277]. Inspired by observations, we used a deep autoencoder architecture for feature-level fusion. Our method involves parallel input of sEMG and IMU data into the deep autoencoder, which processes this multi-modal data to generate a unified dense representation capturing essential features from both sources. This unified representation is then utilized to enhance the accuracy and robustness of handwritten character recognition. This feature fusion aims to leverage spatial, kinematic, and muscle activation information to optimize system performance.

The fundamentals of the autoencoder are detailed in Section 5.3.1, while the modified autoencoder architecture is discussed in this subsection.

We used the basic autoencoder architecture and tuned it according to our problem statement. To accommodate the two input modularity(e.g., IMU and sEMG), each modality has its own encoding subnetwork that independently processes the input data. These encoding subnetworks have separate layers tailored to each modality's characteristics and requirements. To construct a multimodal representation,

each modality begins with its own set of neural layers, which are then followed by a hidden layer that projects the modalities into a shared joint space.

Given two input modalities s_1 (for IMU) and s_2 (for sEMG) and their respective encoder weights and biases W_{enc1}, b_{enc1} and W_{enc2}, b_{enc2}, the two different encoding modules can be represented as:

$$j_1 = f_{enc1}(W_{enc1} \cdot s_1 + b_{enc1}) \tag{6.5}$$

$$j_2 = f_{enc2}(W_{enc2} \cdot s_2 + b_{enc2}) \tag{6.6}$$

Here, j_1 and j_2 are the lower-dimensional latent representations obtained from the IMU and sEMG encoders. The functions f_{enc1} and f_{enc1} are the activation functions applied to the encoded outputs, introducing nonlinearity.

To obtain a joint representation in the common bottleneck layer, the encoded outputs j_1 and j_2 are concatenated as follows:

$$j_{combined} = \text{concat}(j_1, j_2) \tag{6.7}$$

This concatenated vector $j_{combined}$ represents the fused features from both IMU and sEMG modalities.

Given this shared representation, we created separate decoders for both modalities to reconstruct the original inputs from the shared representation. Let the weights and biases for the decoders be W_{dec1}, b_{dec1} for IMU and W_{dec2}, b_{dec2} for sEMG. The two decoding modules can be represented as:

$$s_1{}' = f_{dec1}(W_{dec1} \cdot j_{combined} + b_{dec1}) \tag{6.8}$$

$$s_2{}' = f_{dec2}(W_{dec2} \cdot j_{combined} + b_{dec2}) \tag{6.9}$$

In these equations,$s_1{}'$ and $s_2{}'$ are the reconstructed outputs for IMU and sEMG, respectively. The functions f_{dec1} and f_{dec2} are the activation functions applied to the decoded outputs.

This approach is logical for multi-modal data because it lets the network capture specific features for each modality before establishing a shared representation. From this shared space, the original modalities are reconstructed.

Throughout the training process, the autoencoder aims to minimize the discrepancy between the reconstructed outputs and the original inputs from both modalities, thereby reducing the reconstruction error. The shared bottleneck layer encourages the autoencoder to learn a joint representation that captures the shared and complementary information between the input modalities. The generalized architecture of the proposed model is depicted in Figure 6.5.

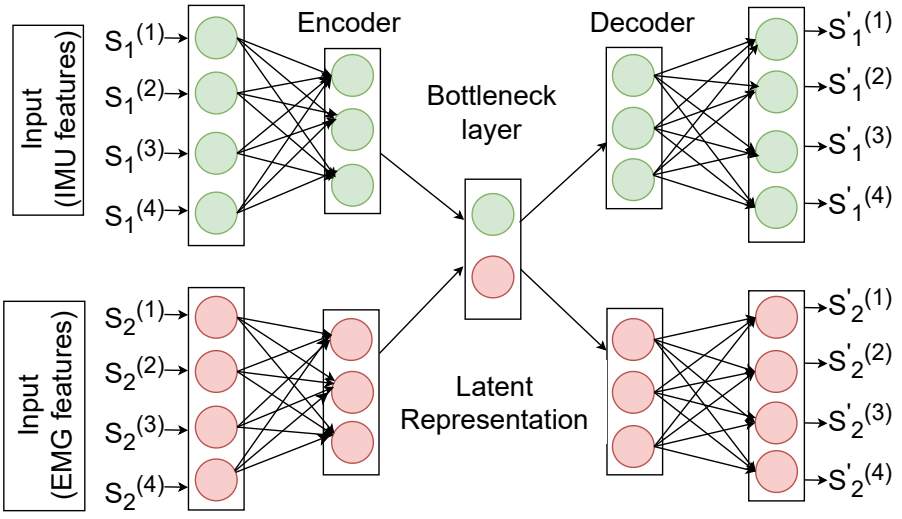

FIGURE 6.5: Demonstration of multi-modal deep auto-encoder architecture

6.3 Results and discussion

Extensive experiments were carried out to determine the effectiveness of the proposed handwriting recognition framework using our collected dataset. The primary objectives of these experiments were as follows:

- Investigating the effectiveness of early feature-level fusion using deep autoencoder architecture for multi-modal sensor data (IMU and sEMG) in whiteboard-based handwriting recognition.
- Comparing the effectiveness of our proposed pipeline with state-of-the-art methods.
- Assessing the performance of our model on IoT-capable low-cost devices (e.g., Raspberry Pi 3).

The whiteboard-based handwritten character recognition problem was framed as a multi-class classification task. The proposed deep multi-modal autoencoder was used to perform the feature fusion. The feature vector derived using the bottleneck layer was further deployed with the classification pipeline. Several standard metrics, including Accuracy, Matthews Correlation Coefficient, Recall, F1 score, and Precision, were employed to assess the performance of the experiments.

6.3.1 Multi-modal deep auto-encoder:Architecture design and implementation details

We conducted extensive experiments to determine the optimal structural parameters for our multi-modal deep autoencoder network, primarily focusing on maximizing classification accuracy for handwriting recognition tasks.

The multimodal deep autoencoder architecture consisted of separate encoding paths for each modality, followed by a fusion layer that combined the encoded representations into a shared representation. For both inputs to encoders, the features

vector was set to 350.

We utilized a deep encoder with five fully connected layers for the IMU module. Each layer employed the Rectified Linear Unit (ReLU) activation function to extract meaningful features from the IMU signals. The sizes of the layers were set to 500,375,250,150,65, respectively, based on the complexity and characteristics of the IMU data. Similarly, a separate deep encoder with five fully connected layers was used for the EMG module, employing the ReLU activation function. The sizes of the EMG layers were also 500, 375, 250, 150, and 65, respectively.

After the last fully connected layer in each encoding path, we incorporated a fusion layer that combined the encoded representations from the IMU and EMG modules. This fusion layer was implemented as a fully connected layer with a size of 130, applying the ReLU activation function. The fused representation then served as the shared representation, capturing joint information from the IMU and EMG modalities.

Moreover, We utilized the Adam optimization algorithm to optimize the multimodal deep autoencoder's performance and enhance the shared representation's quality. Adam optimization is an adaptive algorithm that effectively combines the benefits of adaptive gradient algorithms and momentum-based methods.

During the training process, we employed the Mean Squared Error (MSE) loss function to guide the learning of the multimodal deep autoencoder. This loss function enabled the model to minimize the discrepancy between the reconstructed inputs and the original data samples, thereby encouraging the autoencoder to learn a shared representation that is both concise and informative. By minimizing the MSE loss, we aimed to enhance the overall fidelity and representation quality. Table 6.1 summarizes the architecture with the numbers of parameters.

Component	Layers	Layer Sizes	Activation	Parameters	No. of Parameters
IMU Encoder	5	500, 375, 250, 150, 65	ReLU	Input Vector: 350	504,590
sEMG Encoder	5	500, 375, 250, 150, 65	ReLU	Input Vector: 350	504,590
Fusion Layer	1	130	ReLU	Combines IMU and sEMG Representations	17,030
IMU Decoder	5	65, 150, 250, 375, 500	ReLU	Reconstructs IMU Data	504,875
sEMG Decoder	5	65, 150, 250, 375, 500	ReLU	Reconstructs sEMG Data	504,875
Optimizer	-	-	-	Adam Optimization	-
Loss Function	-	-	-	Mean Squared Error (MSE)	-
Total Parameters					**2,038,960**

TABLE 6.1: Summary of the Multi-Modal Autoencoder Architecture with Parameters

6.3.2 Performance Measures

To rigorously assess the effectiveness and reliability of the proposed method, we implemented a ten-fold cross-validation strategy for evaluating the classification pipeline. We utilized classifiers such as Naive Bayes, Logistic Regression, Linear Discriminant Analysis, Support Vector Machine (SVM), and K-Nearest Neighbor (KNN) to validate the classification results. The learned shared feature representations from the proposed autoencoder were fed into these baseline classifiers for training and validation. These resultant features enabled achieving an average classification accuracy of 99.01%, even with baseline classifiers. The standard performance metrics achieved using the baseline classifiers are presented in Table 6.2. Additionally, Table 6.3 shows the confusion matrix obtained using the SVM classifier on our learned feature representation for the handwritten character recognition (HCR) task. Precision, Recall, and F1 scores for the 26 classes corresponding to different lowercase English alphabets are visualized in Figure 6.6. The model reports the lowest precision, recall, and F1 scores for the letters ‘i’, ‘l’, ‘p’, and ‘s’, indicating difficulty in predicting these characters accurately. In contrast, the model efficiently recognizes the letters ’a’, ’b’, ’c’, ’d’, ’e’, ’f’, ’m’, ’q’, ’t’, ’u’, ’y’, and ’z’, as reflected in their higher precision, recall, and F1 scores.

To further evaluate the predictive capability of our pipeline, we plotted the Receiver Operating Characteristic (ROC) Area Under the Curve (AUC). This curve

assesses the model's ability to discriminate among multiple classes, considering each class's true positive and false positive rates. The model achieved high ROC AUC values for most classes, indicating robust performance in accurately predicting instances across different classes. Figure 6.7 visually demonstrates the efficiency of our model in terms of true positive and false positive rates.

6.3.2.1 Visualization of Learned Representations

The t-SNE plot was used to visually represent the learned feature representation in a lower-dimensional space. By representing the patterns as clusters in the lower dimension, t-SNE ensures that points close to each other in the higher-dimensional space also maintain proximity in the reduced dimension. Figure 6.11 shows the distribution, clustering, and relationships between feature sets corresponding to different classes. The t-SNE plot allows us to examine the relationships between different class features. Features are located close to each other in the plot, suggesting that the auto-encoder has effectively captured the similarities. On the other hand, no clear separation between classes suggests that the auto-encoder struggles to discriminate between different classes and requires further refinement. Figure 6.11 demonstrates that feature sets corresponding to different classes form distinct, minimally overlapping clusters, indicating favorable class separability. This suggests the presence of well-represented class features, facilitating better classification results. Initial observations from the t-SNE plot indicate that the learned feature representation from the multimodal autoencoder network contains sufficient discriminatory properties, enabling classifiers to differentiate between classes easily. However, some data points are clustered with other classes, suggesting difficulty in identifying these points. A higher-dimensional projection may be required to accurately predict their exact characteristics.

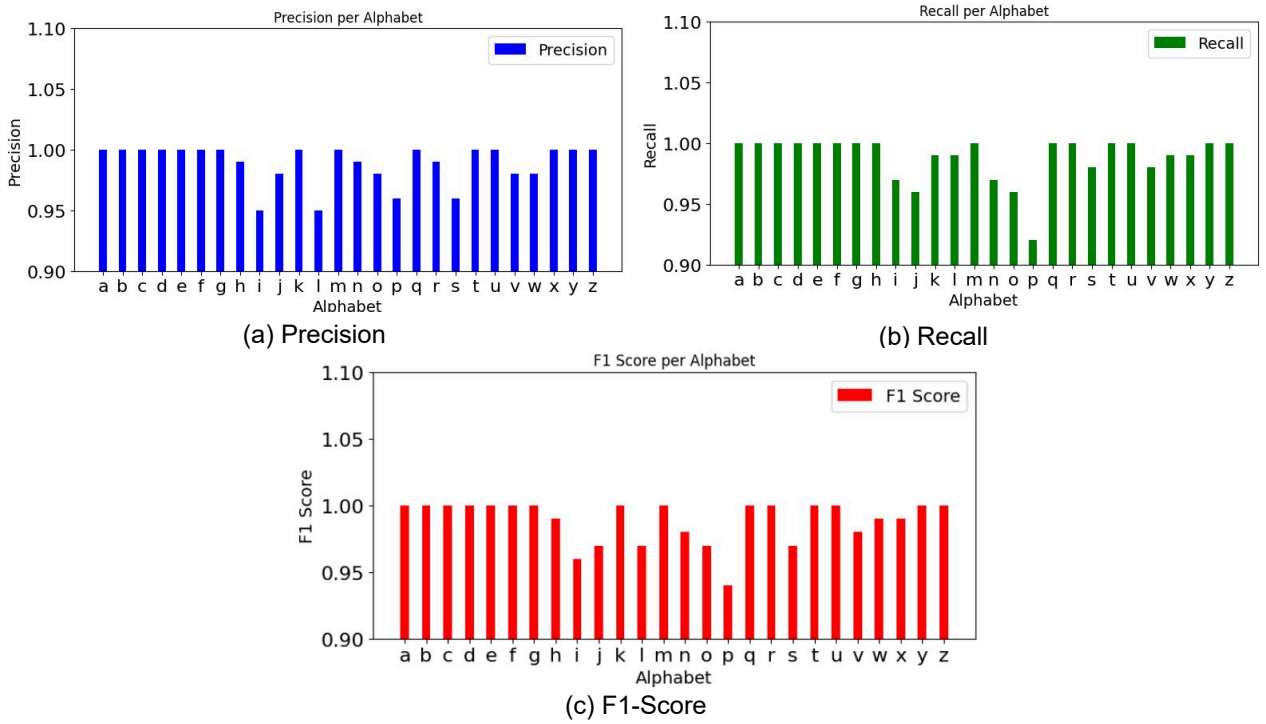

FIGURE 6.6: Precision, Recall, and F1 Score obtained for MMDAE + SVM on MM-HCR dataset

6.3.2.2 Scalability and Performance Analysis

For the MM-HCR dataset, we assessed the scalability and performance of our pipeline by examining the relationships between the learning curve (training and validation scores), model fit times, and the number of training samples. We used various graphs to illustrate these interrelationships.

Figure 6.8 shows the learning curve, where the vertical axis represents the training and validation scores, and the horizontal axis represents the number of training samples. With 2000 samples, the pipeline achieves high training and validation scores above 0.95, smoothly converging toward maximum recognition accuracy as the number of samples increases. Furthermore, as the number of training instances increases, the pipeline consistently refines its performance and steadily converges training and validation scores. This suggests that our pipeline is well-trained with

Classifiers	Accuracy	MCC	Kappa
Naive Bayes	93.90 ± 0.689	92.50	92.44
L-R	98.91 ± 0.107	98.13	98.70
LDA	98.60 ± 0 .234	98.35	98.34
K-nn (K=3)	98.70 ± 0.278	98.05	98.05
K-nn (K=5)	98.70 ± 0 .224	98.08	98.08
SVM (rbf)	99.01 ± 0 .148	98.77	98.97

TABLE 6.2: Different performance metrics achieved using the baseline classifiers

a sufficient number of samples and that adding more training instances does not negatively impact the average recognition accuracy.

Figure 6.9 illustrates the time delay experienced (measured in unit time) during the model training process as a function of the number of training instances. Beyond 10,000 samples, the delay incurred demonstrates an almost linear relation with the number of training samples. A linear increase in training time with the number of training samples suggests that the model scales well with larger datasets.

In contrast, Figure 6.10 portrays the association between the achieved accuracy by the model and the corresponding delay incurred. At around 8,000 samples, the training and validation scores show minimal variation, indicating that the model can effectively use more data and generalize well.

In conclusion, the observations from these Figures (Fig. 6.8, Fig. 6.9, and Fig 6.10) demonstrate the scalability and stability of our pipeline. They suggest that the pipeline can handle additional data without a significant loss in performance, making it suitable for real-world applications involving large datasets.

TABLE 6.3: Confusion matrix obtained for 26 classes using SVM classifier

Class	Precision	Recall	F1 Score	Class	Precision	Recall	F1 Score
a	1.00	1.00	1.00	n	0.99	0.97	0.98
b	1.00	1.00	1.00	o	0.98	0.96	0.97
c	1.00	1.00	1.00	p	0.96	0.92	0.94
d	1.00	1.00	1.00	q	1.00	1.00	1.00
e	1.00	1.00	1.00	r	0.99	1.00	1.00
f	1.00	1.00	1.00	s	0.96	0.98	0.97
g	1.00	1.00	1.00	t	1.00	1.00	1.00
h	0.99	1.00	0.99	u	1.00	1.00	1.00
i	0.95	0.97	0.96	v	0.98	0.98	0.98
j	0.98	0.96	0.97	w	0.98	0.99	0.99
k	1.00	0.99	1.00	x	1.00	0.99	0.99
l	0.95	0.99	0.97	y	1.00	1.00	1.00
m	1.00	1.00	1.00	z	1.00	1.00	1.00

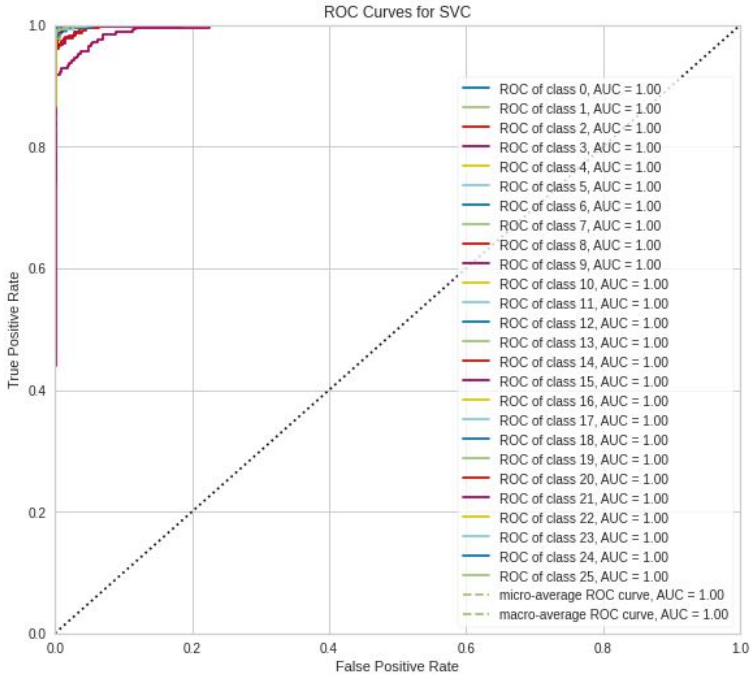

FIGURE 6.7: Plot illustrating the ROC AUC curve obtained for 26 classes using SVM classifier

/

6.3.2.3 Time complexity Analysis

We also tried analyzing the computational costs associated with our proposed model. The computational cost was considered an approximate measure of end-to-end processing delay incurred by our model while classifying a single instance of the dataset. The major components of such processing include delay incurred while data acquisition, feature extraction, feature encoding, and class label prediction using the trained model. The feature encoding step involves generating a shared representation using an autoencoder network. A summary of the calculated delays for our pipeline's major components is presented as

$$T_{Res} = t_{acq} + t_{fea} + t_{enc} + t_{pred} \tag{6.10}$$

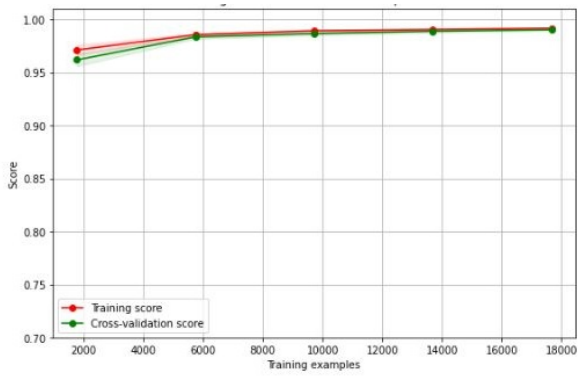

FIGURE 6.8: Training and validation curve obtained using SVM classifier

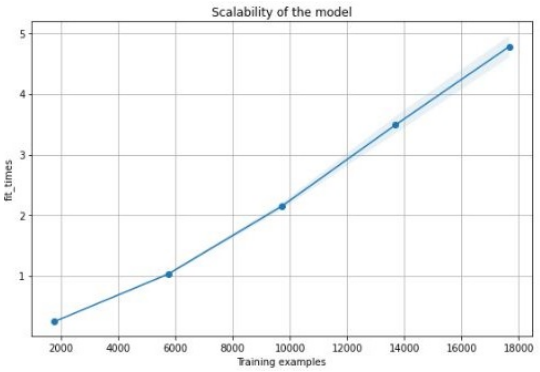

FIGURE 6.9: Plot illustrating the scalability of the model with respect to training samples and time delay (in unit time)

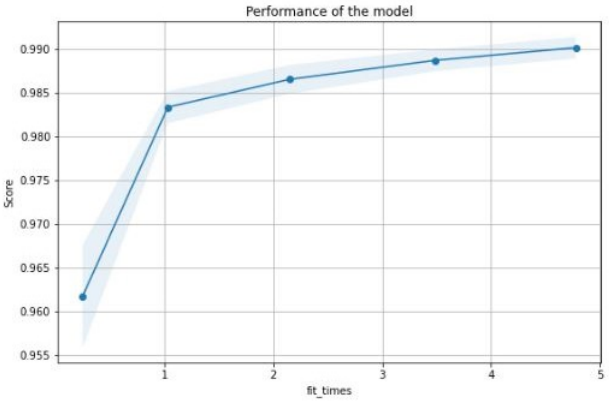

FIGURE 6.10: Plot illustrating the performance of the model with respect to time delay (in unit time)

Where t_{acq} is the raw signal acquisition time while writing a character on the whiteboard, t_{fea} is the time delay incurred while extracting features from IMU and sEMG signals,t_{enc} is the time required to encode the features using a deep autoencoder, and t_{pred} is the time spent by the trained SVM model for predicting a new instance.

We conducted this analysis on a system equipped with an Intel(R) Core(TM) i7-7700 CPU running at 3.60GHz and with 8 GB of RAM. The t_{acq} was taken as 1.1 sec. Optimal t_{enc} or the encoding time by the trained model was obtained as 0.04

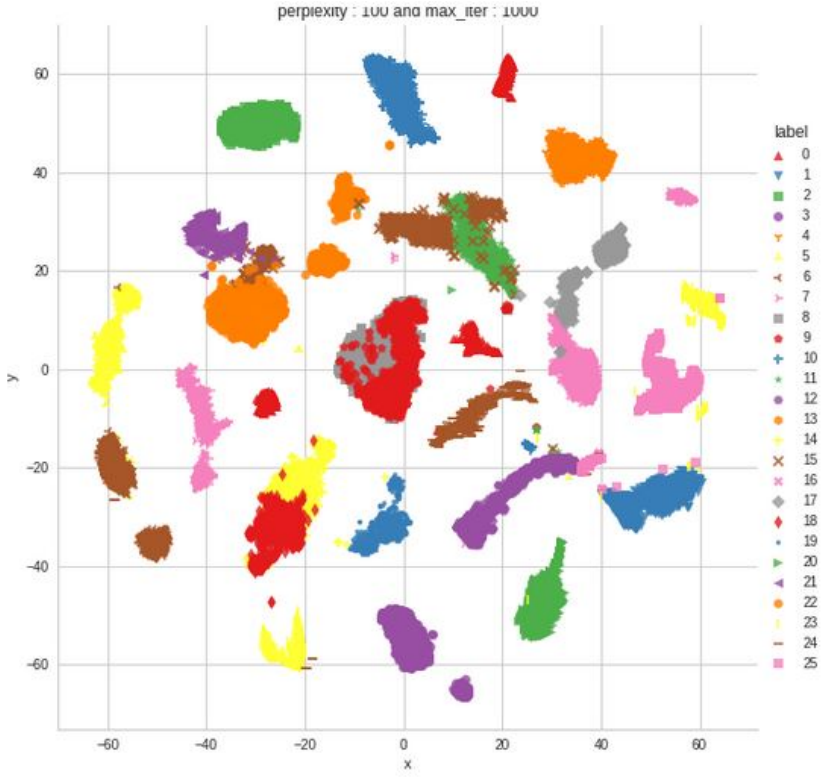

FIGURE 6.11: Plot showing the shared features learned using proposed deep multi-model autoencoder when visualized using t-sne

sec. While the t_{pred} was evaluated as 0.01 sec. t_{fea} is evaluated as the maximum time incurred between extracting 350 features each from sEMG or IMU signals generated for a single character. So, t_{fea} is expressed as

$$t_{fea} = max(Ex_{imu}, Ex_{sEMG}) \tag{6.11}$$

We consider the maximum of the two feature extraction times because our proposed deep auto-encoder architecture strictly requires the two feature vectors to generate the shared representation. The maximum time ensures the availability of both feature vectors for encoding. The Ex_{imu}, Ex_{sEMG} were reported to be 0.0485 and 0.4984 sec when considering an average of over ten executions.

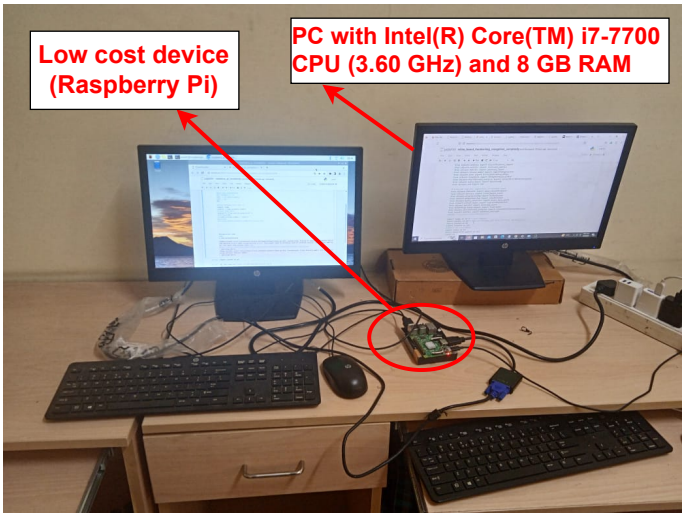

FIGURE 6.12: Experimental setup used to deploy the model on low-cost devices (Raspberry Pi)

Low-Cost Device Performance Evaluation To explore the feasibility of deployment of the proposed framework on IoT-compatible low-cost devices, we ran our model instance on Raspberry Pi 3. While executing on a Raspberry Pi device, the t_{pred} was evaluated as 0.05873 sec compared to 0.0156 sec when deployed on a PC. Figure 6.12 illustrates the experimental setup used to deploy our trained model on low-cost devices. Table 6.4 states the average recognition time while predicting a single whiteboard-based handwritten character.

TABLE 6.4: The average recognition time while predicting a single whiteboard-based handwritten character

t_{acq}	t_{fea}	t_{enc}	t_{pred}	T_{Res} $(t_{acq} + t_{fea} + t_{enc} + t_{pred})$
1.1 sec	0.04 sec	0.04984 sec	0.01 sec	1.19984 sec

6.3.3 Comparison with state-of-the-art approaches

In recent years, significant advancements have been made in gesture and character recognition. In this section, we have selected articles with similar objectives and methodologies to our specific focus on handwritten character recognition and

compared these articles with our proposed approach. These selected articles provide valuable insights into the current state-of-the-art methodologies for character recognition using multimodal sensors.

The comparative evaluation of our approach with other relevant techniques is summarized in Table 6.5. Our proposed method primarily focuses on maximizing recognition accuracy for handwriting recognition tasks, which is crucial for real-time applications. By employing a fusion of SEMG and IMU features at the feature level, we obtained an efficient common feature representation that delivers recognition accuracy comparable to state-of-the-art methods. Initial observations from Table 6.5 indicate that our method has achieved superior average recognition accuracy for various alphabets compared to others.

However, a detailed examination of Table 6.5 reveals that most existing research in this field has relied on conventional sensors such as accelerometers [108], gyroscopes [110], magnetometers [114], and force sensors [107], alongside specialized sensors like ultra-wideband radar [117] and ultrasonic sensors [120], and often in combinations [107, 115]. Notably, no other research has considered the amalgamation of sEMG and IMU sensors for whiteboard-based handwritten character recognition, which presents a novel approach in this domain. The aforementioned articles have demonstrated that accelerometers, gyroscopes, magnetometers, force sensors, and acoustic signals contain essential information that can be utilized for recognizing handwritten characters and gestures with a reasonable degree of accuracy. However, despite the advancements made in these studies, there is still room for improvement. Details of the state-of-the-art methods are given in the related work section.

Those research which has comparable recognition accuracy to ours [112, 115–117, 120] typically involves a limited number of gestures, usually 10-12. Studies that consider a similar number of gestures [107, 113, 114] have demonstrated lower recognition accuracy (around 95%), which is less preferred for developing real-time applications, thereby highlighting potential limitations.

TABLE 6.5: Summary of the related work for multimodal senors-based handwriting recognition tasks

References	Technique used	No of gestures	Sensors used	Device used for data collection	Gestures Modularity	Performance Metrics (Accuracy)
[111]	Feature extraction with ML classifiers	26 alphabets	Accelerometer and gyroscope	Smartwatch	3D gestures made with finger	95%
[112]	Deep learning (BLSTM and BGRU)	12 gestures (digits + alphabets)	Accelerometer and gyroscope	Smartphone	3D gestures	99.15% (alphabets) 99.36% (digits)
[114]	Deep learning 1D-CNN	Digits(0-9) Alphabets(a-z)	Acelerometer+ Gyroscope+ Magnetometer	Fingerworn sensors	In air	95.91%(digits) 93.04% (alphabets)
[107]	Deep learning (CNN, LSTM)	Uppercase alphabets	Acelerometer+ Gyroscope+ Force sensor + Magnetometer	Sensors enhanced pen	On paper	90%
[115]	3D tip estimation + Principal component analysis	Few Handwritten gestures	Acelerometer+ Gyroscope+ Magnetometer	Pen	Regular writing plains	98.2%
[116]	Dynamic Time Wrapping (DTW)	26 alphabets	Acelerometer+ Gyroscope+ Magnetometer	Fingerworn sensors	In air	84.6%
[117]	Deep learning CNN	Digits (0-9)	UWB	Xethra X4-103	In air	98.5%
[119]	Constraint DTW +K-NN	Digits (0-9)	Acelerometer+ Gyroscope+ Magnetometer	Handworn sensor	In air	88.9%
[120]	DTW	Ten gestures	Ultrasonic positioning +IMU	Pen	Unity 3D platform	99.5%
Proposed Approach	Deep learning	26 gestures	sEMG+IMU	Marker	Whiteboard	99.01%

Overall, our model demonstrates promising results for offline recognition of handwritten English alphabets. The experimental outcomes suggest that our deep feature learning approach holds significant potential for online handwriting recognition tasks.

6.3.4 Threats to Validity

Several factors may impact the validity of the proposed model. These are outlined below.

Internal Validity

To ensure consistent data collection, sEMG and IMU sensors were fixed in position on the forearm, and handwriting size was standardized using grid lines. While

these controls improved experimental consistency, they may reduce the model's reliability in less controlled or real-world conditions.

External Validity

The study was limited to right-hand users and isolated character writing on a whiteboard interface. This restricts generalizability to left-handed users, bimanual tasks, or real-world applications like continuous word-level recognition. Future work should test the model across diverse users and settings to broaden its applicability.

Construct Validity

Although the model was trained on dynamic writing tasks that reflect natural movement, it does not account for variations in object interaction, grip force, or task intent. As a result, it may not fully capture the broader concept of functional hand use, limiting alignment with real-world grasp behavior.

Conclusion Validity

The effects of data collected at different times or across sessions were not separately analyzed. These factors may influence model stability over time. Future studies should evaluate this to improve the reliability and reproducibility of the results.

6.4 Summary

In conclusion, this chapter has explored the significant enhancement of dynamic hand gesture recognition through the integration of multi-modal sensor fusion, focusing on sEMG and IMU data. Our research addressed the limitations identified in previous chapters, where dynamic information was lacking when using sEMG signals alone for gesture recognition. By incorporating IMU data, which provides kinematic

insights into motion and orientation, we were able to construct a more comprehensive model capable of effectively capturing the nuances of dynamic gestures.

This chapter presents the architecture of the multi-modal deep autoencoder, which effectively facilitates the generation of a shared representation by capturing and integrating information from both IMU and sEMG modalities. The use of separate encoding paths, each consisting of different fully connected layers, enabled robust feature learning. The architecture successfully captured and integrated features from both IMU and sEMG data, demonstrating its capability to handle multi-modal inputs effectively.

Our experimental results, validated by a new dataset collected explicitly for this study, showed promising accuracy levels comparable to state-of-the-art methods. The MM-HCR dataset was instrumental in training machine learning models that utilize both sEMG and IMU sensor data for recognizing handwritten characters. By leveraging the multi-modal information, the models achieved a more comprehensive understanding of muscle activity and hand motion, leading to improved classification accuracy and robustness in real-world scenarios. This highlights the efficacy of the proposed multi-modal deep autoencoder architecture.

Moreover, this chapter highlighted the model's adaptability to IoT-enabled low-cost devices, such as the Raspberry Pi 3, further showcasing its scalability and practical applicability. By minimizing computational costs and optimizing processing efficiency, our approach ensures that the developed system can be deployed in various environments without compromising performance.

While our model demonstrates significant advancements in dynamic gesture recognition, certain challenges and limitations remain. Future work will focus on addressing sensor placement constraints, expanding the study to include both right and left-hand users, and exploring word-level gesture recognition in real-world applications like smart classrooms. Additionally, examining within-day and inter-day variations will further enhance the model's generalizability and robustness.

In summary, the proposed architecture not only enhances feature learning and integration but also demonstrates significant improvements in classification tasks. Future work could focus on further refining the autoencoder to enhance its discriminative capabilities and exploring its application to other related tasks. This research underscores the potential of multi-modal deep learning approaches in advancing the accuracy and reliability of handwriting recognition systems.

The present work utilizes an input feature vector of length 350, contributing significantly to computational overhead. To reduce this overhead and enhance processing speed, a new metaheuristic-based method is explored in the next chapter.