2024 IEEE International Conference on Systems, Man, and Cybernetics (SMC)
October 6-10, 2024. Sarawak, Malaysia

GRUI: A Novel Gesture Recognition Utilizing UWB Sensor and IMU

Dongjae Lee, Kyeonghyun Yoo, Wooyong Jung, and Hwangnam Kim
Department of Electrical Engineering, Korea University, Seoul, South Korea
{djrandy, seven1705, jy17347, hnkim}@korea.ac.kr

2024 IEEE International Conference on Systems, Man, and Cybernetics (SMC) | 978-1-6654-1020-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/SMC54092.2024.10831691

***Abstract*—In recent advancements in sensor and artificial intelligence technologies, the reliability of gesture recognition has significantly improved, prompting various industrial fields to adopt this technology. However, most gesture recognition systems rely on optical methods because of their high accuracy, despite require complex and computationally intensive processes. Moreover, these systems are associated with high construction costs and are susceptible to environmental factors. This paper introduces a novel gesture recognition system, which effectively tracks and estimates gestures using cost-effective ultra-wideband (UWB) sensors and inertial measurement units (IMU). The system acquires position data of gesture through UWB sensors and includes essential data processing steps such as the detection and removal of abnormal data via IMU, data smoothing with a Kalman filter, and data normalization and scaling. Notably, normalization and scaling are achieved by converting the position data into grayscale images, ensuring the consistency of data features and enhancing gesture recognition accuracy across diverse users. The proposed system employs a convolutional neural network (CNN) model to estimate gestures from these images. Comparative analyses demonstrate that the proposed system exhibits superior gesture classification performance compared to systems utilizing a long-short term memory (LSTM) model and those employing the same CNN model without the aforementioned data processing steps. Therefore, this system is not only cost-effective but also efficiently tracks and estimates gestures, offering significant improvements over existing methods.**

***Index Terms*—Gesture Recognition, Gesture Tracking, Position estimation, Image classification, CNN**

I. Introduction

With the recent advancements in sensor and artificial intelligence technologies, the reliability of gesture recognition has increased by enabling the recognition of more sophisticated and subtle gestures than was previously possible. This technology holds potential for application across various fields [1]. Notably, the automotive industry has incorporated gesture recognition to allow drivers to control infotainment systems, navigation, and other in-vehicle functions, thereby aiming to enhance the driving experience and improve safety by minimizing distractions [2]. Furthermore, gesture recognition is utilized in augmented reality (AR), virtual reality (VR), and the gaming industry, facilitating intuitive and natural interactions that provide users with an immersive experience [3]. Additionally, the COVID-19 pandemic has underscored the importance of reducing physical contact with surfaces in public and shared spaces, positioning gesture recognition technology as a hygienic and touchless alternative for interacting with devices and systems, from elevators to interactive kiosks [4].

For the aforementioned reasons, gesture recognition technology is poised to become increasingly applicable and important. It is largely categorized into optical and non-optical methods, with most devices and systems employing optical gesture recognition technology due to its high accuracy. However, optical methods necessitate high specification hardware to accommodate real-time image processing and gesture recognition, which are complex and computationally intensive processes. Moreover, the requirement for a depth camera, an infrared sensor, and measurement aids contributes to high construction costs. Additionally, being an optical method, it is susceptible to interference from the optical elements of the surrounding environment [5]. In this paper, we propose a novel gesture recognition system that leverages the fusion of UWB sensors and the IMU, which are relatively inexpensive sensors [6]. This system is cost-effective and utilizes UWB sensor positioning, correction with IMU, and a CNN model to estimate gestures, thereby ensuring high gesture recognition accuracy. Furthermore, as it does not rely on optical methods, it is unaffected by the optical elements of the surrounding environment.

Specifically, this paper introduces a novel gesture recognition utilizing UWB sensor and IMU (GRUI). It can accurately position the wearable device. It converts the gesture into an image based on position data, classifies the image using a CNN, and estimates the gesture. The system consists of several main steps. Initially, it involves detecting and removing abnormal data from the position data obtained from the UWB sensor, using the velocity value measured by the IMU. The data processed through this step is then smoothed via a Kalman filter to reduce noise and improve the accuracy of the device's position. Subsequently, the data is converted into a grayscale image that represents the gesture's entire path. Since existing positioning data is time-series data and its features varies for each user even for identical gestures,

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2024-2021-0-01835) supervised by the IITP (Institute of Information Communications Technology Planning Evaluation). Moreover, this work was supported by the National Research Foundation of Korea funded by the Korean Government (grant 2020R1A2C1012389). (Dongjae Lee, and Kyeonghyun Yoo contributed equally to this work.)(Corresponding author: Hwangnam Kim.)

2024 IEEE International Conference on Systems, Man, and Cybernetics (SMC)
October 6-10, 2024. Sarawak, Malaysia

Grui：使用UWB传感器和IMU的一种新颖的手势识别

Dongjae Lee，Kyeonghyun Yoo，Wooyong Jung和Hwangnam Kim电气工程系，韩国韩国首尔，韩国首尔{ djrandy，Sequenn1705，Jy17347

Abstract - 在传感器和人工智能技术方面的最新进步中，手势识别的可靠性已大大提高，促使各种工业领域采用这项技术。然而，尽管需要复杂且计算密集的过程，但大多数手势识别系统依赖于光学方法。此外，这些系统与高建筑成本有关，并且容易受到环境因素的影响。本文介绍了一种新型的手势识别系统，该系统使用具有成本效益的超宽带（UWB）传感器和惯性测量单元（IMU）有效地跟踪和估计手势。该系统通过UWB传感器获取手势的位置数据，并包括基本的数据处理步骤，例如通过IMU检测和删除异常数据，使用卡尔曼过滤器进行数据平滑以及数据归一化和缩放。值得注意的是，通过将位置数据转换为灰度图像，确保数据功能的一致性并增强各种用户的手势识别精度来实现归一化和缩放。拟议的系统采用卷积神经网络（CNN）模型来估计这些图像的手势。比较分析表明，与使用长期术语内存（LSTM）模型的系统相比，所提出的系统表现出了出色的手势分类性能，并且使用了相同的CNN模型的系统，而没有上述数据处理步骤。因此，该系统不仅具有成本效益，而且有效地跟踪和估计手势，从而对现有方法提供了重大改进。

Index Terms - 最佳识别，手势跟踪，位置估计，图像分类，CNN

I.简介

随着传感器和人工intel-ligence技术的最新进步，手势识别的可靠性通过实现比以前更复杂和更微妙的手势的识别来提高。该技术具有在各个领域的应用[1]的潜力。值得注意的是，汽车行业已将手势识别纳入了允许驾驶员控制信息娱乐系统，导航和其他车辆内部功能，从而旨在通过最大程度地减少分心来增强驾驶体验并提高安全性[2]。此外，手势识别用于增强现实（AR），虚拟现实（VR）以及游戏行业，促进了直观和自然的互动，从而为用户提供沉浸式体验[3]。此外，COVID-19大流行强调了减少与公共场所和共享空间中表面的物理接触的重要性，将手势识别技术定位为与设备和系统相互作用的卫生和无触摸替代方案，从电梯到交互式售货亭[4]。

出于上述原因，手势识别技术有望变得越来越适用和重要。它主要分为光学和非光学方法，大多数设备和系统由于其较高的准确性而采用光姿势识别技术。但是，光学方法需要高规范硬件来适应复杂且计算密集型过程的实时图像处理和手势识别。此外，对深度摄像头，红外传感器和测量辅助设备的需求有助于高建筑成本。另外，作为一种光学方法，它容易受到周围环境的光学元素的干扰[5]。在本文中，我们提出了一个新型的手势识别系统，该系统利用UWB传感器和IMU的融合，它们是相对便宜的传感器[6]。该系统具有成本效益，并利用UWB传感器定位，使用IMU进行校正以及CNN模型来估计手势，从而确保了高手势识别的准确性。此外，由于它不依赖于光学方法，因此不受周围环境的光学元素的影响。

具体而言，本文使用UWB传感器和IMU（GRUI）引入了一种新颖的手势识别。它可以准确定位可穿戴设备。它将手势转换为基于位置数据的图像，使用CNN对图像进行分类，并估算手势。该系统由几个主要步骤组成。最初，它涉及使用IMU测得的速度值从UWB传感器获得的位置数据中检测和删除异常数据。然后，通过Kalman滤光片平滑通过此步骤处理的数据，以降低噪声并提高设备位置的准确性。随后，数据将转换为代表手势整个路径的灰度图像。由于现有的定位数据是时间序列数据，并且它的功能对于每个用户都会有所不同，即使是相同的手势，

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2024-2021-0-01835) supervised by the IITP (Institute of Information Communications Technology Planning Evaluation). Moreover, this work was supported by the National Research Foundation of Korea funded by the Korean Government (grant 2020R1A2C1012389). (Dongjae Lee, and Kyeonghyun Yoo contributed equally to this work.)(Corresponding author: Hwangnam Kim.)

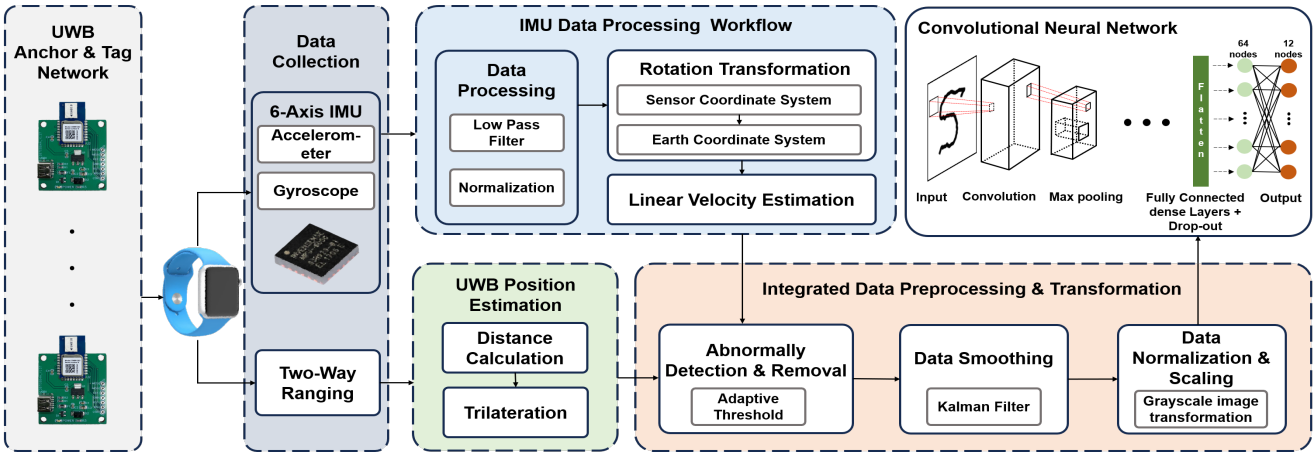

Fig. 1. Overview of GRUI system Structure

normalization and scaling processes are typically required for each user. However, by converting this data into a grayscale image, these processes can be omitted, capturing consistent features within the same type of gesture for all users. This approach ensures data consistency and allows for gesture analysis on the same scale for all users. Moreover, since this image does not require color information, it reduces the data size and amount of calculation by applying grayscale. Finally, gestures are estimated by inputting the grayscale image into a CNN which is well-suited for image classification [7]. CNNs are powerful machine learning algorithms that automatically learn and classify image data features, enabling the accurate recognition and classification of various user gestures. This system facilitates gesture tracking and estimation, allowing for the recognition of user's gestures in the automotive industry, AR, VR, gaming industry, and public and shared spaces, and their use as commands.The contributions of this system are as follows:

1) This paper presents a sensor fusion-based data processing and analysis system that utilizes UWB sensors and IMU to accurately estimate the position of wearable devices and then converts the position data into images to effectively classify users' gestures using CNN.
2) We compare the performance of gesture classification using a CNN model with that of a LSTM model to evaluate how converting to a grayscale image affects gesture classification performance. In addition, systems similar to GRUI, except for the data correction step through IMU and Kalman filters, are compared with GRUI to evaluate the effectiveness of data correction using IMU and Kalman filters in gesture classification. These performance comparisons provide important evidence of the proposed method's effectiveness and superiority over existing methods.

The remaining part of this paper is organized as follows. Section II provides an in-depth look at the GRUI architecture, detailing the data collection process, and the integration of UWB and IMU data. In Section III, we delve into the experiment setup and database description, followed by the training and evaluation of the proposed system. Finally, Section IV concludes our paper, summarizing the key findings and contributions.

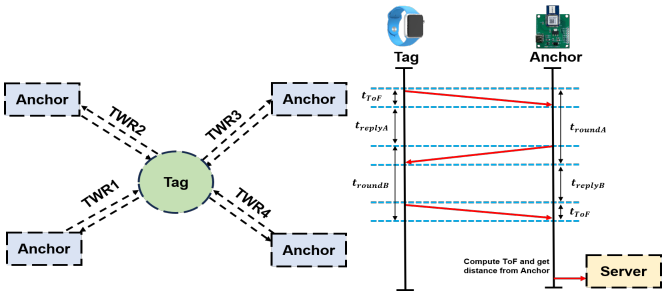

Fig. 2. Diagram of UWB Layout and Communication Protocol

II. GRUI Architecture

The overall structure of the proposed system is shown in Fig. 1. The UWB sensors and IMU collect data. In each module, distance obtained from trilateration using the UWB sensors and velocity calculated through the IMU, are transmitted to the server and processed. The server obtains a distance value from data obtained from the UWB sensor and a maximum velocity value after completion of the gesture from IMU data. Thereafter, abnormal UWB position values are removed by applying a threshold value, which is derived from the maximum velocity value of the IMU, to the UWB position data. The removed data is smoothed through the Kalman filter to reduce noise and improve the accuracy of the data. The smoothed data is transformed into a grayscale image, normalized, and inputted into the CNN model for training and classification. Details are in Sections II-A1 and II-A2.

A. Data Collection

This section provides a technical description of how UWB and IMU data are collected when the user performs gestures

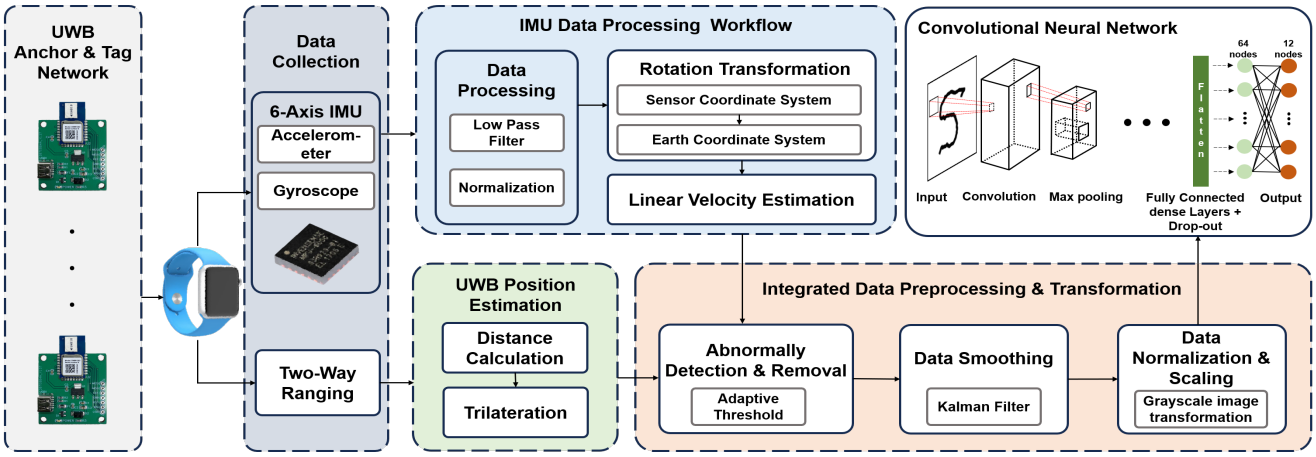

图1。GRUI系统结构的概述

每个用户通常需要归一化和缩放过程。但是，通过将这些数据转换为灰度图像，可以省略这些过程，从而为所有用户捕获相同类型的手势中的一致功能。此方法确保数据一致性，并允许所有用户对同一量表进行手势分析。此外，由于此图像不需要颜色信息，因此可以通过应用灰度来减少数据大小和计算量。最后，通过将灰度图像输入到CNN中来估计手势，该图像非常适合图像分类[7]。 CNN是强大的机器学习算法，它们会自动学习和分类图像数据功能，从而可以准确识别和分类各种用户手势。该系统促进了手势跟踪和估计，允许识别汽车行业，AR，VR，游戏行业以及公共和共享空间的用户手势，并用作命令。该系统的贡献如下：如下：

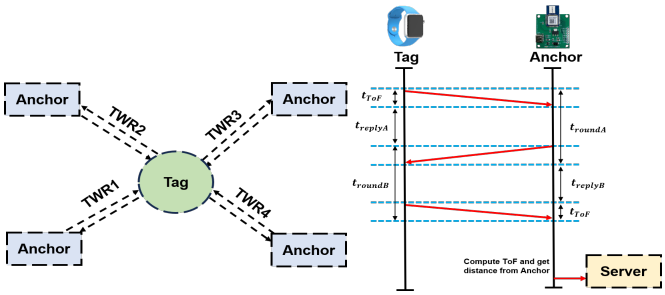

图2。UWB布局和通信协议的图

实验设置和数据库描述，然后进行培训和评估。最后，第四节结束了我们的论文，总结了关键发现和贡献。

1）本文提出了一种基于传感器融合的数据处理和分析系统，该系统利用UWB传感器和IMU准确估算可穿戴设备的位置，然后将位置数据转换为图像，以使用CNN有效地对用户的手势进行分类。 2）我们使用CNN模型与LSTM模型的手势分类的性能进行比较，以评估转换为灰度图像的如何影响手势分类的性能。此外，将类似于GRUI的系统（除了通过IMU和Kalman过滤器的数据校正步骤）与GRUI进行了比较，以评估使用IMU和Kalman过滤器在手势分类中使用IMU和Kalman过滤器的数据校正的有效性。这些性能比较为拟议方法的有效性和优越性比现有方法的优势提供了重要的影响。

本文的其余部分如下组织。第二节对GRUI架构进行了深入的了解，详细介绍了数据收集过程以及UWB和IMU数据的集成。在第三节中，我们深入研究

ii。 Grui架构

所提出的系统的整体结构如图1所示。UWB传感器和IMU收集数据。在每个模块中，使用UWB传感器和通过IMU计算的速度获得的三重速度获得的距离被传输到服务器并处理。该服务器从UWB传感器获得的数据中获得距离值，并在IMU数据完成手势后获得最大速度值。此后，通过应用一个从IMU的最大速度值派生到UWB位置数据的阈值来消除异常的UWB位置值。删除的数据通过卡尔曼过滤器平滑，以减少噪声并提高数据的准确性。平滑的数据被转换为灰度图像，并将其归一化，并将其输入到CNN模型中进行训练和分类。详细信息在II-A1和II-A2节中。

A. Data Collection

本节提供了有关在用户执行手势时如何收集UWB和IMU数据的技术描述

on wearable device. In addition, it provides a basic understanding of UWB trilateration and IMU velocity calculation methods.

1) UWB Position Estimation: In the proposed system, UWB technology plays a crucial role in accurately determining the position of the wearable device attached to users. This section provides an in-depth explanation of the UWB position estimation process. The layout of the UWB sensor consists of setting four anchors and one tag as shown in Fig. 2, the anchor is fixed in a known position, and the tag is attached to the wearable device. The position estimating process begins with the start of the TWR (Two-Way Ranging) protocol shown in the right part of Fig. 2, which is essential for measuring the ToF of the signal exchanged between the anchor and the tag. The TWR protocol consists of multiple message exchanges, and the protocol begins with the sending of a poll message from the tag to the anchor. After that, a response message is transmitted from the anchor, and finally, a final message is transmitted from the tag to the anchor. The time interval between the transmission and reception of these messages is accurately measured, which, combined with the speed of light, is used to accurately measure the distance between the tag and each anchor. The following equation describes how the distance between one anchor and one tag is calculated using the Time of Flight (ToF) and the speed of light:

$$\begin{aligned} t_{\text{ToF}} &= \frac{t_{\text{roundA}} \times t_{\text{roundB}} - t_{\text{replyA}} \times t_{\text{replyB}}}{t_{\text{roundA}} + t_{\text{roundB}} + t_{\text{replyA}} + t_{\text{replyB}}}, \\ d &= c \times t_{\text{ToF}}. \end{aligned} \tag{1}$$

In the provided formula, t_{ToF} denotes the ToF, which is the calculated duration taken by a signal to traverse from the sending to the receiving device and back. The variables t_{roundA} and t_{roundB} signify the first and second measurements of the round-trip time, respectively, from device A to device B and back to device A. On the other hand, t_{replyA} and t_{replyB} represent the first and second measurements of the reply times from device A and device B, respectively. The c stands for the speed of light, denoting the propagation speed of the signal in the medium. Lastly, d is the distance between the two devices.

When each anchor sends distance data to the server via the TWR protocol, the server estimates the tag's position using the trilateration formula. Below is the trilateration formula:

$$\begin{aligned} A &= -2x_1 + 2x_2, \\ B &= -2y_1 + 2y_2, \\ C &= d_1^2 - d_2^2 - x_1^2 - y_1^2 + x_2^2 + y_2^2, \\ D &= -2x_2 + 2x_3, \\ E &= -2y_2 + 2y_3, \\ F &= d_2^2 - d_3^2 - x_2^2 - y_2^2 + x_3^2 + y_3^2, \\ x &= \frac{CE - FB}{EA - BD}, \\ y &= \frac{CD - AF}{BD - AE}, \end{aligned} \tag{2}$$

where x and y are the target position coordinates, $(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)$ are the coordinates of the four anchors, and d_1, d_2, d_3, d_4 are the distances from each anchor to the target. With known anchor coordinates and distances, the unknown x and y can be resolved.

When position estimating is performed using a UWB sensor alone, communication failures may occur due to multiple paths of reflection, refraction, or scattering of radio waves caused by the surrounding environment, and the performance of the position estimating system such as time synchronization errors, delay times, noise and interference is deteriorated. In this paper, after calculating the velocity value using IMU, it is used as a threshold on the server to remove abnormal data among UWB position data. The method of obtaining the velocity value of the IMU is described in II-A2.

2) IMU Velocity Estimation: At the data collection stage, data from the gyroscope and accelerometer are obtained from the IMU, which is timestamp, angular velocity, and acceleration. The sensor calculates the posture of the wearable device, that is, the rotation matrix, at each time by applying the AHRS (Attitude and Heading Reference System) algorithm called Mahony. The AHRS algorithm uses gyroscope and accelerometer data to calculate the sensor's rotation matrix, enabling transformation from the sensor coordinate system to the Earth coordinate system. Conversion to the Earth's coordinate system yields continuous stable results. The transformation process between coordinate systems is defined as follows:

$$\mathbf{ECS} = \mathbf{M} \times \mathbf{SCS}, \tag{3}$$

where ECS is the vector of coordinates in the Earth Coordinate System, M is the transformation matrix that encapsulates the rotation, translation, and possibly scaling required to map SCS to ECS. SCS is the vector of coordinates in the Sensor Coordinate System.

The linear acceleration is obtained by removing the Earth's gravitational acceleration from the transformed Earth coordinate system acceleration data. This process removes the effects of gravity while maintaining only information about the sensor's linear gesture. The linear velocity of the sensor is then estimated by multiplying the linear acceleration by the sensor's sampling interval, and the maximum linear velocity value is extracted and transmitted to the server during the gesture measurement. The process of calculating linear acceleration and linear velocity is defined as follows.

B. Integrated Data Preprocessing and Transformation

Position data extraction from UWB sensors provides an important dataset for capturing user gesture dynamic feature. However, since this data is time series data, sophisticated preprocessing is essential to effectively utilize the embedded information. This section describes the systematic process undertaken to refine and transform UWB position data into a format suitable for gesture classification.

1) Abnormal Data Detection and Removal: In order to detect and remove abnormal values from position data estimated from UWB sensors, the adaptive threshold technique

在可穿戴设备上。此外，它提供了UWB三级征服和IMU速度计算方法的基本不明。

1) UWB Position Estimation: 在拟议的系统中，UWB技术在准确确定用户附加可穿戴设备的位置方面起着至关重要的作用。本节提供了UWB位置估计过程的深入说明。 UWB传感器的布局包括设置四个锚和一个标签，如图2所示，锚位于已知位置，标签连接到可穿戴设备上。位置估计过程始于图2右侧的TWR（双向范围）协议的开始，这对于测量锚和标签之间交换的信号的TOF至关重要。 TWR协议由多个消息交换组成，该协议是从将标签发送到锚点的投票消息开始。之后，从锚点传输了响应消息，最后，将最终消息从标签传输到锚。这些消息的传输和接收之间的时间间隔是准确测量的，结合光速，用于准确测量标签和每个锚之间的距离。以下等式描述了一个锚和一个标签之间的距离是如何使用飞行时间（TOF）和光速计算的：

$$\begin{aligned} t_{\text{ToF}} &= \frac{t_{\text{roundA}} \times t_{\text{roundB}} - t_{\text{replyA}} \times t_{\text{replyB}}}{t_{\text{roundA}} + t_{\text{roundB}} + t_{\text{replyA}} + t_{\text{replyB}}}, \\ d &= c \times t_{\text{ToF}}. \end{aligned} \tag{1}$$

在提供的公式中，t_{ToF}表示TOF，这是信号从发送到接收器件和后背的信号所计算出的持续时间。变量t_{roundA}和t_{roundB}分别表示往返时间的第一个和第二个测量值，分别从设备A到设备B，然后返回设备A。另一方面，t_{replyA}和t_{replyB}分别代表来自设备A和设备B的回复时间的第一个和第二个测量值。 c代表光速，表示信号在介质中的传播速度。最后，d是两个设备之间的距离。

当每个锚通过TWR协议将距离数据发送到服务器时，服务器使用三级配方估算标签的位置。以下是三材料公式：

$$\begin{aligned} A &= -2x_1 + 2x_2, \\ B &= -2y_1 + 2y_2, \\ C &= d_1^2 - d_2^2 - x_1^2 - y_1^2 + x_2^2 + y_2^2, \\ D &= -2x_2 + 2x_3, \\ E &= -2y_2 + 2y_3, \\ F &= d_2^2 - d_3^2 - x_2^2 - y_2^2 + x_3^2 + y_3^2, \\ x &= \frac{CE - FB}{EA - BD}, \\ y &= \frac{CD - AF}{BD - AE}, \end{aligned} \tag{2}$$

其中x和y是目标位置坐标，(x_1, y_1)，$(x_2, y_2), (x_3, y_3), (x_4, y_4)$是四个锚的坐标，$d_1, d_2, d_3, d_4$是从每个锚到目标的距离。有了已知的锚坐标和距离，可以解决未知的x和y。

当单独使用UWB传感器进行位置估计时，可能会由于周围环境引起的多个反射，折射或散射的多个路径而发生通信故障，以及位置估计系统的性能，例如时间同步误差，延迟时间，噪声，噪声和干扰。在本文中，在使用IMU计算速度值后，它被用作服务器上的阈值，以删除UWB位置数据之间的异常数据。 II-A2中描述了获得IMU速度值的方法。

2) IMU Velocity Estimation: 在数据收集阶段，陀螺仪和加速度计的数据是从IMU获得的，IMU是时间戳，角速度和加速度。传感器通过应用称为Mahony的AHR（态度和标题参考系统）算法来计算每次可穿戴设备的姿势，即旋转矩阵。 AHRS算法使用陀螺仪和加速度计数据来计算传感器的旋转矩阵，从而使从传感器坐标系转换为接地坐标系。转换为地球坐标系统会产生连续的稳定结果。坐标系之间的转移过程定义如下：

$$\mathbf{ECS} = \mathbf{M} \times \mathbf{SCS}, \tag{3}$$

其中ECS是地球坐标系统中坐标的向量，M是将旋转，翻译和可能比例映射到SCS中所需的缩放所需的转换矩阵。 SCS是传感器坐标系中坐标的向量。

线性加速度是通过从转化的地球辅助系统加速度数据中去除地球引力加速度来获得的。该过程消除了重力的影响，同时仅维持有关传感器线性手势的信息。然后，通过将线性加速度乘以传感器的采样间隔来估算传感器的线性速度，并在手势测量过程中提取并将最大线性速度值提取并传输到服务器。计算线性加速度和线性速度的过程定义如下。

B. Integrated Data Preprocessing and Transformation

从UWB传感器提取位置数据提供了一个重要的数据集，用于捕获用户手势动态功能。但是，由于此数据是时间序列数据，因此复杂的预处理对于有效利用嵌入式信息至关重要。本节介绍了为了完善和将UWB位置数据转换为适合手势分类的格式而进行的系统过程。

1) Abnormal Data Detection and Removal: 为了从UWB传感器估算的位置数据中检测和删除异常值，自适应阈值技术

using the maximum linear velocity value estimated by IMU is applied. The adaptive threshold technique is a method of dynamically setting thresholds in consideration of data volatility. Data with greater velocity variability than threshold are considered abnormal data and removed by comparing UWB position data from the previous time and position data from the next time from each time of UWB position data. After applying this method, the quality of the data is improved and the accuracy of gesture recognition was improved.

2) Data Smoothing: Kalman filters are widely used for real-time data analysis and uncertainty minimization [8]. This is very useful for estimating the current state of the system from the noise-mixed measurement data. In this study, after detecting and removing abnormal position data, the data was smoothed by applying a Kalman filter to the remaining data. This process reduces data noise and provides more accurate and reliable data for future data normalization and scaling steps. The application of the Kalman filter reduces the variability of refined UWB position data, thereby enabling more accurate position estimation. The smoothing process is based on the mathematical model of the Kalman filter, which describes the dynamic behavior of the system and considers measurement noise and process noise. Measurement noise represents the uncertainty or error in the measurement data, while process noise accounts for the uncertainty in the system dynamics. The Kalman filter is defined as follows:

$$\begin{aligned} \hat{x}_{k|k-1} &= A\hat{x}_{k-1|k-1} + Bu_k, \\ P_{k|k-1} &= AP_{k-1|k-1}A^T + Q, \\ K_k &= P_{k|k-1}H^T(HP_{k|k-1}H^T + R)^{-1}, \\ \hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k(z_k - H\hat{x}_{k|k-1}), \\ P_{k|k} &= (I - K_kH)P_{k|k-1}, \end{aligned} \tag{4}$$

where the matrices Q and R represent the covariances of the system and measurement noises respectively. Through experimental analysis, the values for the system noise covariance Q and the measurement noise covariance R were determined to be 0.0001 and 0.1 respectively. The state transition and observation models, represented by matrices A and H respectively, along with the control input model B, are crucial in determining the evolution of the system state and the relationship between the true state and the observations. Through the iterative application of these equations, the Kalman filter provides estimates of the system state $\hat{x}_{k|k}$ and updates the state and measurement covariance matrices $P_{k|k}$ and $P_{k|k-1}$, facilitating accurate state estimation in the presence of noisy measurements and system uncertainties.

C. Data Normalization and Scaling

The gesture data obtained through the previous process is time series data, and due to the characteristics of each user and the diversity of movements, even if the same type of gesture is expressed, it may appear differently for each user. This data variability can pose challenges for machine learning models aiming to accurately classify gestures, so a normalization and scaling process of data is essential to address this. The data normalization and scaling process is the process of adjusting the values of each data point to be within a specific range, which makes the training process of machine learning models more stable and efficient.

Algorithm 1 Gesture Classification from Fusion Positioning

Inputs: X, Y, $velocity_{max}$, dt
Output: $gesture$
Initialize:
$P_r, P_f \leftarrow []$
while True **do**
  for $i = 1$ to length(X) **do**
    $dis \leftarrow \sqrt{(X[i] - X[i-1])^2 + (Y[i] - Y[i-1])^2}$
    $velocity \leftarrow dis/dt$
    if $speed < velocity_{max}$ **then**
      $P_r.append([X[i], Y[i]])$
    end if
  end for
  for $i = 0$ to length(P_r) **do**
    $P_{KF} \leftarrow$ KF$(P_r[i])$
    $P_f.append(P_{KF})$
  end for
  $grayscale\_image \leftarrow pos2gray(P_f)$
  $gesture \leftarrow$ CNN$(grayscale_image)$
  Reset the values of P_f and P_r to empty lists
end while

In this study, refined position data is utilized by applying the aforementioned adaptive threshold method. This refined data is transformed into a grayscale, and this transformation process maps each pixel value of the position data to a value between 0 and 255 to create a grayscale image with a pixel size of 159×158. At this time, by setting the image to grayscale, unnecessary color information is eliminated, enabling efficient data management and calculation. The grayscale image converted into this process is input to the machine learning model to perform classification of gestures. Grayscale image conversion makes the training process of machine learning models more efficient and accurate by applying normalization and scaling effects to data, ultimately contributing to the improvement of gesture recognition performance. The grayscale image, generated through this process, is then used as an input for the CNN to perform gesture recognition and classification. Specific aspects of machine learning and CNN model construction and training are detailed in III-A.

Algorithm 1 illustrates all the steps executed on the server in the gesture recognition process of GRUI as previously described. Discussing the parameters, X and Y represent the arrays of tag position data obtained from the UWB sensor during the gesture classification execution time, and $velocity_{\max}$ represents the maximum velocity value computed from the IMU. KF denotes the linear Kalman filter, while *pos2gray* is a function that transforms position data into grayscale image data. Lastly, dis represents the Euclidean distance values

使用IMU估计的最大线性速度值。自适应阈值技术是一种考虑数据波动率的动态设置阈值的方法。比阈值更大的速度可变性的数据被视为异常数据，并通过比较上一次UWB位置数据的每个时间的上一次时间和位置数据的UWB位置数据来删除。应用此方法后，数据的质量得到提高，并提高了手势识别的准确性。

2) Data Smoothing: 卡尔曼过滤器被广泛用于实时数据分析和不确定性最小化[8]。这对于从噪声混合测量数据估算系统的当前状态非常有用。在这项研究中，在检测和删除异常位置数据后，通过将卡尔曼过滤器应用于其余数据来平滑数据。此过程减少了数据噪声，并为将来的数据归一化和缩放步骤提供了更准确和可靠的数据。Kalman滤波器的应用降低了精致的UWB位置数据的可变性，从而实现了更准确的位置估计。平滑过程基于卡尔曼滤波器的数学模型，该模型描述了系统的动态行为并考虑了测量噪声和过程噪声。测量噪声代表测量数据中的不确定性或错误，而过程噪声则说明了系统动力学的不确定性。卡尔曼过滤器定义如下：

$$\begin{aligned} \hat{x}_{k|k-1} &= A\hat{x}_{k-1|k-1} + Bu_k, \\ P_{k|k-1} &= AP_{k-1|k-1}A^T + Q, \\ K_k &= P_{k|k-1}H^T(HP_{k|k-1}H^T + R)^{-1}, \\ \hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k(z_k - H\hat{x}_{k|k-1}), \\ P_{k|k} &= (I - K_kH)P_{k|k-1}, \end{aligned} \tag{4}$$

矩阵Q和R分别表示系统的协方差和测量噪声。通过实验分析，分别确定了系统噪声协调Q和测量噪声协方差R的值分别为0 . 0001和0 . 1。状态过渡和观察模型分别由矩阵A和H表示，以及控制输入模型B，对于确定系统状态的演变以及真实状态与观测值之间的关系至关重要。通过这些方程式的迭代应用，卡尔曼过滤器提供了系统状态$\hat{x}_{k|k}$的估计值，并更新状态和测量协方差矩阵$P_{k|k}$和$P_{k|k-1}$，从而在存在嘈杂的测量和系统的不认定的情况下促进了准确的状态估计。

C. Data Normalization and Scaling

通过上一个过程获得的手势数据是时间序列数据，并且由于每个用户的特征和运动的多样性，即使表达了相同类型的手势，每个用户的表现也可能不同。此数据可变性可能会对旨在准确分类手势的机器学习模型构成挑战，因此，数据的归一化和缩放过程对于

Algorithm 1 Gesture Classification from Fusion Positioning

Inputs: X, Y, $velocity_{max}$, dt
Output: $gesture$
Initialize:
$P_r, P_f \leftarrow []$
while True **do**
  for $i = 1$ to length(X) **do**
    $dis \leftarrow \sqrt{(X[i] - X[i-1])^2 + (Y[i] - Y[i-1])^2}$
    $velocity \leftarrow dis/dt$
    if $speed < velocity_{max}$ **then**
      $P_r.append([X[i], Y[i]])$
    end if
  end for
  for $i = 0$ to length(P_r) **do**
    $P_{KF} \leftarrow$ KF$(P_r[i])$
    $P_f.append(P_{KF})$
  end for
  $grayscale\_image \leftarrow pos2gray(P_f)$
  $gesture \leftarrow$ CNN$(grayscale_image)$
  Reset the values of P_f and P_r to empty lists
end while

解决这个问题。数据归一化和缩放过程是调整每个数据点的值在特定范围内的过程，这使机器学习模型的训练过程更加稳定和高效。

在这项研究中，通过应用上述自适应阈值方法来利用精致的位置数据。此精制数据被转换为灰度，并且此转换过程将位置数据的每个像素值映射到0到255之间的每个像素值，以创建灰度图像，像素大小为159 × 158。此时，通过将图像设置为灰度，不必要的颜色信息可以消除，从而消除了不成功的颜色信息，从而实现了富有帮助的数据管理和计算。转换为此过程的灰度图像输入到机器学习模型以执行手势分类。灰度图像转换使机器学习模型的训练过程通过将归一化和缩放效果应用于数据，更加有效，准确，最终有助于改善手势识别的能力。然后，通过此过程生成的灰度图像被用作CNN执行手势识别和分类的输入。III-A中详细介绍了机器学习和CNN模型构建和培训的特定方面。

算法1说明了以前撰写的GRUI手势识别过程中服务器上执行的所有步骤。讨论参数X和Y表示从UWB传感器获得手势分类执行时间的标签位置数据的AR-RARS，而$velocity_{\max}$表示从IMU计算的最大速度值。KF表示线性kalman滤波器，而*pos2gray*是将位置数据转换为灰度图像数据的函数。最后，dis表示欧几里得距离值

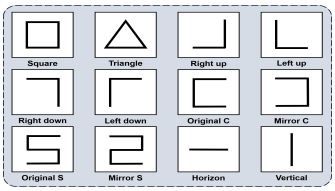

Fig. 3. Shape of 12 gestures

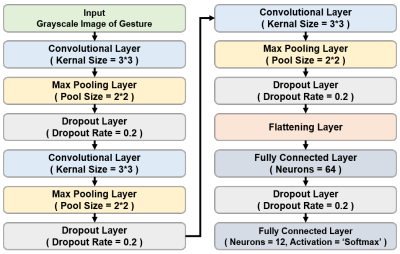

Fig. 4. Schematic Diagram of the CNN Structure

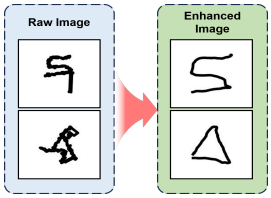

Fig. 5. Data Preprocessing Procedure

between UWB position time-series data within the loop. The process can be summarized as follows: Utilizing $velocity_{\max}$ as the threshold, abnormal position values among the UWB position data are removed. Subsequently, the position data is smoothed using KF, transformed into grayscale image data, and then inputted into the CNN model to recognize the gesture. Finally, P_f and P_s are initialized as empty arrays for recognizing the next gesture.

III. Training and Evaluation

A. Experiment Setup

We used the DWM1000 UWB sensor from Qorvo and the MPU6050 IMU, which is a 6-axis sensor with a built-in accelerometer and gyroscope. The anchor and tag used Raspberry Pi 4B 4GB. Anchors were placed indoors in a square shape of about $2m \times 2m$, and gestures were carried out in various ways in a square range with a side length between 0.45m and 0.6m. The server used a laptop with an Intel Core i7-7700HQ (2.8 GHz) and NVIDIA GeForce GTX 1060 GPU.

B. Database Description

In Fig. 3, there are a total of 12 kinds of gestures. It is easy to detect and classify when it is composed of gestures with distinctly different features, but we also included gestures with similar features to confirm the exact performance of our system. The database has 1200 gray scale images, each 159×158 pixels. 900 were used for training, 300 for testing.

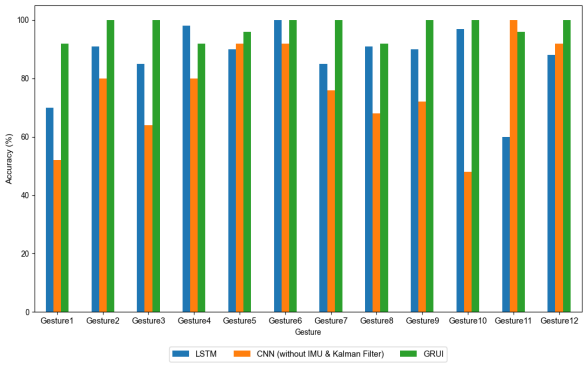

Fig. 6. Gesture Classification Accuracy Comparison Between Systems

C. Training

We used a CNN training model consisting of a total of 13 layers, including three convolution layers, three max Pooling layers, four dropout layers, one flattening layer, and two dense layers, as shown in Fig. 4. The image used as the input is a single channel shape of 159×158 pixels because it reduces the data size by removing unnecessary color information with grayscale. The following is a description of the overall structure of the CNN model. The first layer, the convolution layer, is utilized by applying the ReLU activation function to 16 3×3 size filters. In addition, the vanishing gradient problem may be prevented by using the ReLU activation function. In the case of the second layer, max pooling layer, a 2×2 sized pool is used. Reduce overfitting by applying a dropout rate of 0.2 in the last layer, dropout layer. It goes through this process three times in total. In this case, the number of filters of the convolution layer is 16, 32, and 64, respectively. After going through the process described above, the result from the above process is made into a 1D vector through the flattening layer. After that, it reaches the final dense layer through a dense layer using 64 neurons and a ReLU activation function and a dropout layer with a dropout rate of 0.2. In the final dense layer, classification operations can be performed through 12 neurons and the Softmax activation function. The model was trained with a batch size of 8 for 10 epochs. The training dataset is 900 images, 75% of the database.

D. Evaluation

In this subsection, we evaluate the effectiveness and efficiency of the proposed approach. To ensure the reliability of our results, we categorized our evaluation into two main components, Data Preprocessing and Performance Analys. First, we describe the preprocessing taken in each process to make the data suitable for our CNN model. Subsequently, we evaluated the impact of the grayscale transformation on gesture classification performance by comparing the classification capabilities of the CNN and LSTM used in the proposed GRUI. Furthermore, we assessed the effectiveness of these methods by comparing GRUI with CNN models trained on

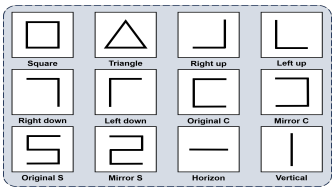

图3。12个手势的形状

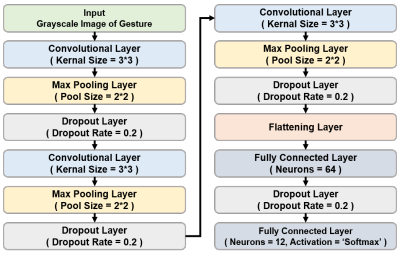

图4。CNN结构的示意图

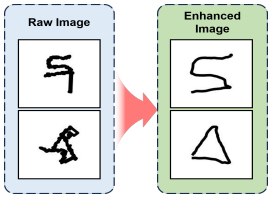

图5。数据预处理程序

在UWB位置循环中的时间序列数据之间。该过程可以汇总如下：使用$velocity_{\max}$作为阈值，UWB位置数据中的异常位置值。随后，使用KF平滑位置数据，转换为灰度图像数据，然后输入到CNN模型中以识别手势。最后，P_f和P_s被初始化为空数组，以识别下一个手势。

iii。培训和评估

A. Experiment Setup

我们使用了Qorvo和MPU6050 IMU的DWM1000 UWB传感器，该传感器是一个具有加速度计和陀螺仪内置的6轴传感器。锚和标签使用Raspberry Pi 4B 4GB。将锚定在大约2 $m \times 2\ m$的平方形室内放置，并以各种方式在平方范围内进行手势，侧面长度在0.45m至0.6m之间。该服务器使用具有Intel Core i7-7700HQ（2.8 GHz）和NVIDIA GEFORCE GTX 1060 GPU的笔记本电脑。

B. Database Description

在图3中，总共有12种手势。当它由具有明显不同特征的手势组成时，很容易检测和分类，但是我们还提供了具有相似功能的手势以确认系统的确切性能。该数据库具有1200个灰度图像，每个图像159 × 158像素。 900用于训练，300用于测试。

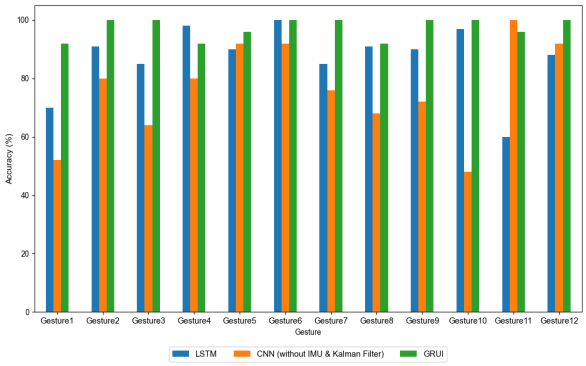

图6。系统之间的手势分类精度比较

C. Training

我们使用了CNN训练模型，该模型由总共13层组成，包括三层卷积层，三个最大池层，四个辍学层，一层扁平层和两个密集的层，如图4所示。输入使用的图像是一个单个通道形状的159 × 158 pixels的单个通道形状，因为它通过将数据大小降低，因为它通过将数据大小降低了，并且通过不正确的色彩来降低数据大小。以下是对CNN模型的整体结构的描述。第一层是卷积层，通过将Relu激活函数应用于16 3 × 3大小过滤器来使用。另外，可以使用Relu激活函数来预防消失的梯度问题。在第二层的情况下，最大池层，使用2 × 2个大小的池。通过在最后一层的辍学层中施加0.2的辍学率来减少过度拟合。它总共经历了三次。在这种情况下，卷积层的过滤器数分别为16、32和64。经过上述过程之后，上述过程的结果通过扁平层制成1D矢量。之后，它使用64个神经元和一个relu激活函数和辍学率为0.2的辍学层通过致密层达到最终密集层。在最终密集层中，可以通过12个神经元和软磁性激活函数进行分类操作。该模型的批量大小为8个时期的批量训练。培训数据集为900张图像，占数据库的75%。

D. Evaluation

在本小节中，我们评估了拟议方法的有效性和效率。为了确保结果的可靠性，我们将评估分为两个主要组成部分，即数据预处理和性能分析。首先，我们描述了每个过程中所做的预处理，以使数据适合我们的CNN模型。随后，我们通过比较拟议的GRUI中使用的CNN和LSTM的分类能力来评估灰度转化对GESTURE分类性能的影响。此外，我们通过将GRUI与受过训练的CNN模型进行比较来评估这些方法的有效性

images without the utilization of adaptive thresholding and data smoothing.

TABLE I
EXPERIMENTAL RESULTS OF DIFFERENT SYSTEMS AND MODELS

	LSTM	**CNN without IMU & Kalman Filter**	**GRUI**
Input Data	Time Series	Image	Image
Precision	0.886	0.792	0.978
Recall	0.871	0.763	0.977
F1-Score	0.873	0.766	0.977

1) Data Preprocessing: Data quality has a decisive impact on the performance of machine learning models. In this paper, we underwent three preprocessing processes to enhance the quality of position data obtained from four UWB anchors. In the first, we applied the maximum velocity measured by the IMU as a threshold to the 2D X and Y position data obtained from trilateration with the four UWB anchors. This method utilizes the maximum velocity measured by the IMU to automatically set the threshold appropriately for various users. Secondly, to increase the continuity and consistency of the data, a linear Kalman filter was applied to smooth out the data. Kalman filter was instrumental in reducing noise or minor fluctuations in the data and clarifying the overall flow of the data. The results of these processes are depicted in Fig 5. Lastly, to overcome the issue of varying data sizes across different individuals and gesture types, position data was transformed into grayscale. Through this transformation, the data was normalized and scaled. As a result, a consistent input format was secured for all users and gesture types, contributing to the training of the CNN model and the accuracy of gesture classification. These three preprocessing processes played a pivotal role in enhancing the performance of our CNN model and significantly contributed to ensuring data quality in dynamic environments, such as wearable devices.

2) Performance Analysis: LSTM is used as a comparative benchmark for GRUI, and the raw time series position data obtained from sensors were trained by applying the adaptive threshold method and the Kalman filter. Additionally, the CNN was trained from time series data converted into images without the adaptive threshold method which utilize IMU, and the data smoothing process using the Kalman filter. This CNN based approach is employed as a comparative indicator for GRUI. The test dataset consists of 300 images, representing 25% of the entire database, with 25 images for each gesture. We employed evaluation metrics including Accuracy, Precision, Recall, and F1-score. The reported accuracy for GRUI ranged from 92% to 100%, while LSTM achieved a range of 60% to 100%, and the CNN based approach achieved a range of 48% to 100%. As depicted in Fig. 6, GRUI consistently demonstrates the highest accuracy in classifying various gestures. Table I displays the detailed experimental results for the entire test dataset across different systems and models. It is evident that GRUI outperforms all others across all metrics. Therefore, converting time series data of gestures into images and applying CNN leads to more accurate gesture recognition than using time series data with LSTM. This advantage is particularly pronounced when individualization is required, encompassing normalization and scaling, to accommodate diverse people, gesture types, and sizes. Furthermore, the application of the adaptive threshold method and data smoothing processes significantly enhances the accuracy of gesture recognition.

IV. CONCLUSION

This paper has explored a gesture recognition system designed to meet the rapidly growing demands of various fields, propelled by advancements in sensor and artificial intelligence technologies. To address this, we introduced GRUI, a novel gesture recognition system that utilizes data fusion from UWB sensors and IMU to accurately determine the position of device. This position data is then transformed into a grayscale image, which facilitates real-time user gesture classification using a CNN. Furthermore, we assessed the gesture classification performance by comparing the CNN model utilized in GRUI with the LSTM model and evaluated the influence of grayscale image transformation on gesture classification performance.

The findings of this research significantly contribute to the field of gesture recognition by demonstrating an alternative to optical methods. By leveraging cost-effective UWB sensors and IMU, GRUI accurately recognizes gestures. Moreover, GRUI's ability to monitor and analyze user movements in a variety of wearable applications and situations showcases its potential applicability across diverse fields, including the automotive industry, AR, VR, the gaming industry, and public and shared spaces.

REFERENCES

[1] R. Li, Z. Liu, and J. Tan, "A survey on 3d hand pose estimation: Cameras, methods, and datasets," *Pattern Recognition*, vol. 93, pp. 251–272, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320319301724

[2] L. Khaleghi, U. Artan, A. Etemad, and J. A. Marshall, "Touchless control of heavy equipment using low-cost hand gesture recognition," *IEEE Internet of Things Magazine*, vol. 5, no. 1, pp. 54–57, 2022.

[3] B. K. Chakraborty, D. Sarma, M. K. Bhuyan, and K. F. MacDorman, "Review of constraints on vision-based gesture recognition for human–computer interaction," *IET Computer Vision*, vol. 12, no. 1, pp. 3–15, 2018.

[4] T. Antes, L. G. De Oliveira, A. Diewald, E. Bekker, A. Bhutani, and T. Zwick, "Velocities in human hand gestures for radar-based gesture recognition applications," in *2023 IEEE Radar Conference (RadarConf23)*. IEEE, 2023, pp. 1–5.

[5] S. Hazra and A. Santra, "Robust gesture recognition using millimetric-wave radar system," *IEEE Sensors Letters*, vol. 2, no. 4, pp. 1–4, 2018.

[6] S. Lee, S. Yoo, J. Y. Lee, S. Park, and H. Kim, "Drone positioning system using uwb sensing and out-of-band control," *IEEE Sensors Journal*, vol. 22, no. 6, pp. 5329–5343, 2022.

[7] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, and T. Chen, "Recent advances in convolutional neural networks," *Pattern Recognition*, vol. 77, pp. 354–377, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320317304120

[8] K. Sung and H. Kim, "Simplified kf-based energy-efficient vehicle positioning for smartphones," *Journal of Communications and Networks*, vol. 22, no. 2, pp. 93–107, 2020.

不利用自适应阈值和数据平滑的图像。

表I不同系统和模型的实验结果

	LSTM	CNN without IMU & Kalman Filter	GRUI
Input Data	Time Series	Image	Image
Precision	0.886	0.792	0.978
Recall	0.871	0.763	0.977
F1-Score	0.873	0.766	0.977

1) Data Preprocessing: 数据质量对机器学习模型的性能有决定性的影响。在本文中，我们进行了三个预处理过程，以增强从四个UWB锚点获得的位置数据的质量。首先，我们应用了由IMU测量的最大速度作为阈值对2D *X*和*Y*的位置数据，该数据从具有四个UWB锚固剂获得的三材料中获得。此方法利用IMU测量的最大速度可以自动为各种用户设置阈值。其次，为了提高数据的连续性和一致性，应用了线性的卡尔曼滤波器来平滑数据。卡尔曼过滤器在减少数据中的噪声或微小波动方面发挥了作用，并阐明了数据的整体流量。这些过程的结果如图5所示。最后，为了克服不同个体和手势类型的不同数据大小的问题，位置数据已转化为灰度。通过这种转换，数据归一化并缩放。结果，为所有用户和手势类型确保了一致的输入格式，这有助于训练CNN模型和手势分类的准确性。这三个预处理过程在增强CNN模型的性能方面发挥了关键作用，并有助于确保在动态环境（例如可穿戴设备）中的数据质量。

2) Performance Analysis: LSTM用作GRUI的比较基准，并且通过应用自适应阈值方法和Kalman滤波器训练了从传感器获得的原始时间序列位置数据。此外，通过使用IMU的自适应阈值方法转换为图像的时间序列数据，对CNN进行了训练，并使用Kalman滤波器进行了数据平滑过程。这种基于CNN的方法被用作GRUI的比较指标。测试数据集由300张图像组成，代表了整个数据库的25%，每个手势都有25张图像。我们采用了评估指标，包括准确性，精度，召回和F1得分。报道的GRUI准确性范围从92%到100%，而LSTM的范围为60%至100%，基于CNN的方法的范围为48%至100%。如图6所示，Grui始终证明了分类各种手势的最高准确性。表I显示了不同系统和模型的整个测试数据集的详细实验结果。显然，格鲁伊的表现要优于所有其他所有指标。因此，将手势的时间序列数据转换为图像并应用CNN会导致与使用LSTM使用时间序列数据更准确的手势识别。当需要进行分歧，包括标准化和缩放，以适应多元化的人，手势类型和大小时，这种优势尤其明显。此外，适应性阈值方法和数据平滑过程的应用显着提高了手势识别的准确性。

iv。结论

本文探讨了一种旨在满足各个领域的快速增长需求的手势识别系统，这是由传感器和人工知识技术的进步所推动的。为了解决这个问题，我们引入了Grui，这是一种新型的手势识别系统，该系统利用UWB传感器和IMU的数据融合来准确确定设备的位置。然后将该位置数据转换为灰度图像，该图像有助于使用CNN实时用户手势分类。此外，我们通过将GRUI中使用的CNN模型与LSTM模型进行比较，评估了手势分类性能，并评估了灰度图像转换对手势分类性能的影响。

这项研究的发现通过证明光学方法的替代方案，显着促进了手势识别领域。通过利用具有成本效益的UWB传感器和IMU，Grui可以准确地识别手势。此外，Grui能够监视和分析各种可穿戴应用程序和情况的用户运动能力展示了其在包括汽车行业，AR，VR，游戏行业以及公共和共享空间在内的各种领域的潜在适用性。

参考

[1] R. Li，Z. Liu和J. Tan，“对3D手姿势估计的调查：相机，方法和数据集”，” *Pattern Recognition*，第1卷。 93，第251–272页，2019年。[在线]。可用：https：//www.sciendirect.com/science/article/pii/s0031320319301724 [2] 5，不。 1，第54-57、2022页。[3] B. K. Chakraborty，D。Sarma，M。K。Bhuyan和K. F. MacDorman，“对基于视力的人类互动的手势识别的限制的评论，” 12，不。 1，第3–15页，2018年。[4] T. Antes，L。G. de Oliveira，A。Diewald，E。Bekker，A。Bhutani和T. Zwick，“人体手势的基于雷达基于雷达的手势的手势识别应用程序的速度”IEEE，2023年，第1-5页。 [5] S. hazra和A. Santra，“使用毫米波雷达系统的强大手势识别”，*IEEE Sensors Letters*，第1卷。 2，不。 4，第1-4页，2018年。[6] S. Lee，S。Yoo，J。Y. Lee，S。Park和H. Kim，“使用UWB感应和带外控制的无人机定位系统”，*IEEE Sensors Journal*，第1卷。 22，否。 6，第5329–5343、2022页。[7] J. Gu，Z. Wang，J。Kuen，L。Ma，A。Shahroudy，B。Shuai，T。Liu，T。Liu，X。Wang，G。Wang，G。Wang，J。Cai，J。Cai和T. Chen，“最近的互惠神经网络的进步，” 77，第354–377页，2018年。[在线]。可用：https：//www.sciendirect.com/science/article/pii/s0031320317304120 [8] K. Sung and H. 22，否。 2，第93–107页，2020年。