Enhancing Dynamic Hand Gesture Recognition using Feature Concatenation via Multi-Input Hybrid Model

Original Scientific paper

Djazila Souhila Korti
Belhadj Bouchaib University of Ain-Temouchent
Smart Structures Laboratory (SSL)
Faculty of Technology, Department of Telecommunication
Ain-Temouchent, Algeria
souhila.korti@univ-temouchent.edu.dz

Zohra Slimane
Abou Bekr Belkaid University of Tlemcen
Faculty of Technology, Department of Telecommunication
Tlemcen, Algeria
zoh_slimani@yahoo.fr

Kheira Lakhdari
Abou Bekr Belkaid University of Tlemcen
Faculty of Technology, Department of Telecommunication
Tlemcen, Algeria
kblakhdari@gmail.com

Abstract *– Radar-based hand gesture recognition is an important research area that provides suitable support for various applications, such as human-computer interaction and healthcare monitoring. Several deep learning algorithms for gesture recognition using Impulse Radio Ultra-Wide Band (IR-UWB) have been proposed. Most of them focus on achieving high performance, which requires a huge amount of data. The procedure of acquiring and annotating data remains a complex, costly, and time-consuming task. Moreover, processing a large volume of data usually requires a complex model with very large training parameters, high computation, and memory consumption. To overcome these shortcomings, we propose a simple data processing approach along with a lightweight multi-input hybrid model structure to enhance performance. We aim to improve the existing state-of-the-art results obtained using an available IR-UWB gesture dataset consisting of range-time images of dynamic hand gestures. First, these images are extended using the Sobel filter, which generates low-level feature representations for each sample. These represent the gradient images in the x-direction, the y-direction, and both the x- and y-directions. Next, we apply these representations as inputs to a three-input Convolutional Neural Network- Long Short-Term Memory- Support Vector Machine (CNN-LSTM-SVM) model. Each one is provided to a separate CNN branch and then concatenated for further processing by the LSTM. This combination allows for the automatic extraction of richer spatiotemporal features of the target with no manual engineering approach or prior domain knowledge. To select the optimal classifier for our model and achieve a high recognition rate, the SVM hyperparameters are tuned using the Optuna framework. Our proposed multi-input hybrid model achieved high performance on several parameters, including 98.27% accuracy, 98.30% precision, 98.29% recall, and 98.27% F1-score while ensuring low complexity. Experimental results indicate that the proposed approach improves accuracy and prevents the model from overfitting.*

Keywords*: hand gesture recognition, IR-UWB, data expansion, multi-input, CNN-LSTM, feature concatenation, multi-class SVM, Optuna*

1. INTRODUCTION

Hand Gesture Recognition (HGR) is a very important research area that provides adequate support for several applications such as human-computer interaction and healthcare monitoring [1,2]. A significant effort has been devoted to gesture recognition using different sensing technologies [3]. Conventional HGR approaches mainly use wearable and optical sensors. These frameworks are highly accurate but represent several drawbacks. Wearable sensors such as gloves require carrying a load of cables that connect the device to a computer while per-

forming the gesture [4]. This makes the system impractical and can cause discomfort for users. In contrast, optical sensors such as cameras do not require any devices attached to the body [5]. However, one prominent concern is the risk of privacy violation when used in personal settings. Furthermore, there arise some situations wherein gesture recognition via cameras is difficult such as sudden lighting changes and the presence of severe occlusions. To overcome the above shortcomings, radar-based sensing systems are proposed [6]. Impulse Radio Ultra-Wide Band (IR-UWB) has recently emerged as one of the most effective and promising non-contact sensors for HGR. It has been deployed in a network fashion for HGR to develop applications such as control car devices [7], wireless keyboards [8,9], and sign language-based communication systems [10,11]. The IR-UWB has the advantage of being remotely operable in a non-intrusive manner. It does not capture any visual images which allow the users to feel unrestrained. Furthermore, the IR-UWB offers an inexpensive and robust system that operates with low power consumption and performs well in both highly lit and dark environments. In addition, it completely avoids the problem of occlusion owing to its high penetration capabilities through obstacles and walls.

The HGR process involves extracting a set of relevant features from the sensor data that best describe a gesture [6], allowing it to be identified with a high recognition rate regardless of the environment in which it is performed or the person performing it [12].

Based on the existing feature extraction techniques, an HGR system can be classified as a traditional or deep model [13]. A traditional model relies on hand-crafted feature extraction, which requires pre-processing of the data to reduce dimensionality and determine appropriate features [14]. Several methods have been investigated for IR-UWB-based HGR, including Multi-Layer Perceptron (MLP) [15], SVM [16], K-Nearest Neighbors (KNN) [17], and K-means [18]. Although these approaches have managed to achieve impressive recognition rates, they are not straightforward, requiring a lot of work for manual feature extraction that heavily depends on human experience and domain knowledge [19]. On the other hand, a deep model eliminates the manual feature extraction phase by replacing it with automatic processing where multi-level features are automatically extracted from raw data, involving less human intervention [7].

In terms of deep models, Convolutional Neural Network (CNN) [20][21] and Recurrent Neural Network (RNN) [22] are the most prominent approaches used for HGR. CNN is considered one of the most efficient deep models for image classification tasks [23]. It acts as a spatial feature extractor and allows one to learn high-level representations in a hierarchical manner using a set of stacked convolutional layers. Several neural network methods have been analyzed, and the results show that CNN is effective for classifying image data generated by IR-UWB for HGR [24-26]. As for RNN, it is used to analyze sequential data. The most commonly deployed variant of RNN is the Long-Short-Term Memory model (LSTM) [27]. This model is designed with a memory mechanism that uses gates, allowing for exploiting and learning relevant temporal patterns in data. LSTM has been effectively used in various HGR studies with radar [28], and it has also proven to boost performance in terms of classification accuracy when used in a hybrid configuration with CNN [29].

Training deep models from scratch typically requires a large amount of data, which is often not available. Acquiring and annotating remote sensing data can be complex, laborious, and time-consuming, making it challenging to gather the necessary amount of data. As a result, the concept of transfer learning was introduced [30], which involves reusing a previously trained model developed for one task in a new task. However, transfer learning-based algorithms may exhibit unpredictable performance if there is a mismatch between the source and target learning content. Another approach is to transform radar data into different domains, where useful features can be extracted and fused for classification [6,31]. However, this approach requires significant processing and computational resources.

This paper proposes a simple processing approach based on low-level feature extraction to increase the number of samples, as well as a multi-input hybrid model to improve the existing results on an available real-world dataset of dynamic hand gestures acquired using an IR-UWB. Three major points have been addressed in this work. First, the limited amount of data used to train a model. A gesture may not be fully described by a single representation; hence the need to extend the data. The introduction of our data processing using the Sobel filter to extract gradient features significantly filters out unnecessary information while retaining the main features. This process can not only increase the amount of information used to describe a target but also enhance the bottom features by reducing the noise in the data and providing more diversified information. Second, we proposed a three-input CNN-LSTM feature extractor that takes advantage of automatic domain-aware extraction and concatenation of complementary features from the same target to provide more exhaustive spatiotemporal information. Third, the model combines the strength of CNN-LSTM and SVM to improve the recognition accuracy and generalization ability while maintaining a simple architecture with a reasonable number of parameters.

The major contributions of this paper are:

- Using preprocessing steps to extend the amount of data in each class to prevent overfitting.
- Providing a lightweight three-input architecture to process the input data, resulting in considerable improvement in training time.
- Utilising CNN-LSTM layers for automatic spatio-temporal feature learning without any manual en-

gineering or prior domain knowledge.

- Using a multi-class SVM classifier for efficient classification.
- Achieving a high recognition rate and outperforming current state-of-the-art models used for IR-UWB-based hand gesture recognition.

The rest of the paper is organized as follows: Section 2 provides a brief review of recent scholarly works related to HGR-based IR-UWB radar. Section 3 presents a description of the proposed system, including dataset processing and model implementation. Section 4 provides the experimental results and comparative performance of the proposed model. The discussion is presented in Section 5, and Section 6 concludes the paper.

2. RELATED WORKS

2.1. TRADITIONAL MODELS

In the HGR process, a gesture needs to be represented by a suitable set of features. Ghaffar *et al.* [19] used the Histogram of Oriented Gradient (HOG) to extract features from the data of 4 IR-UWB. The resulting features were merged and fed as input to an SVM, resulting in an accuracy of 96% for the classification of 4 gestures. Li *et al.* [17] tested several combinations of Cumulative Distribution Density (CDD) features extracted from IR-UWB spectrograms to train a KNN algorithm, with the highest accuracy achieved being about 82.4%. Khan *et al.* [18] extracted three features, namely the variance of the Probability Density Function (PDF) of the magnitude histogram, Time Of Arrival (TOA) variation, and frequency from the data of an IR-UWB. They used the K-means clustering algorithm to classify 5 gestures, achieving an accuracy of 98%.

2.2. DEEP MODELS

2.2.1. CNN

Ahmed *et al.* [32] proposed a four-layer CNN and tested it on the dataset used in this paper. The task was to automatically extract features from range-time radar spectrograms and classify 12 dynamic hand gestures. A recognition accuracy of 94% was achieved. In another study, Ahmed *et al.* [7] proposed a system to recognize gestures for controlling electronic devices inside a car. The authors used a six-layer CNN to extract features from range-time radar spectrograms converted into grayscale images. The system achieved an accuracy of 96% in recognizing the 5 gestures used in the study. Khan *et al.* [33] employed a five-layer CNN to classify hand gestures based on image trajectory patterns generated from multiple IR-UWB.

2.2.2. LSTM

Noori *et al.* [34] investigated an LSTM architecture tested on the same dataset used in this paper. Their model showed superior performance compared to [32] by reaching an accuracy of 97%. However, it had 847,055 trainable parameters, making it computationally heavy. Park *et al.* [35] proposed a recognition algorithm based on LSTM for the classification of 6 dynamic hand gestures and achieved an accuracy of 90.5%.

2.2.3. CNN-LSTM

Skaria *et al.* [36] implemented a hybrid model that combines CNN and LSTM layers, trained on a 3D tensor of stacked range-Doppler frames. The CNN-LSTM achieved a high accuracy of 96.15%. In another work by the same authors, two sensors were investigated for robust gesture classification, namely an IR-UWB and a thermal sensor [37]. CNN-LSTM layers were employed on both radar and thermal signals, achieving an accuracy of 99% for 14 hand gestures.

3. MATERIALS AND METHOD

3.1. DATASET

This study aims to improve the existing results on the public UWB Gestures dataset proposed by Ahmed *et al.* [32]. The dataset consists of 12 classes of dynamic hand gestures, namely left-right (LR), right-left (RL), up-down(UD), down-up (DU), diag-LR-UD, diag-LR-DU, diag-RL-UD, diag-RL-DU, clockwise, counter-clockwise, push-in, and empty gestures. Each gesture is performed 100 times by 8 volunteers and acquired using three XeThru X4 IR-UWBs (Fig. 1). In order to compare the performance of our three-input CNN-LSTM-SVM against other models using the same dataset, we followed the same procedure and used only the left radar data, which consists of 9,600 range-time images.

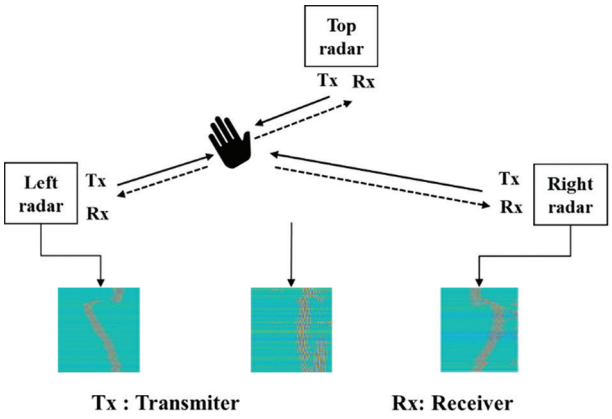

Fig. 1. Collection of the UWB-Gestures dataset using three UWB radars.

3.2. DATA PROCESSING

A major problem in training deep models from scratch is the huge amount of data required. Using a small dataset usually results in overfitting and decreased model performance. To achieve high generalization capability, we propose extending the dataset and generating low-level image features to use as an in-

dependent input for the model. These features include the image gradient in the x-direction, y-direction, and both x and y-directions generated by the Sobel filter. First, the samples are resized to 75x75 to decrease the computational cost, then converted to binary images by applying thresholding. Next, the Sobel filter is applied in the horizontal and vertical directions to calculate the gradient in the x-direction (Dx) and y-direction (Dy) for each sample. Finally, the gradient images for both x and y-directions are generated by taking the square root of the sum of Dx and Dy for each sample. By doing this, the size of the dataset is tripled, and a total of 25,200 images are generated without creating duplicates (Fig. 2).

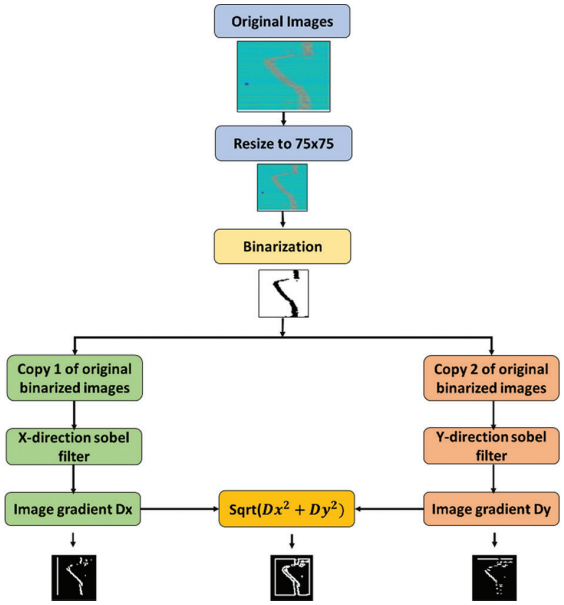

Fig. 2. Flow graph of data processing.

3.3. METHOD

The proposed approach is designed to classify dynamic hand gestures, which are formed by a consecutive sequence of poses where their characteristics vary over time. Therefore, the recognition process of dynamic gestures takes into account both spatial and temporal information. To enable the analysis of spatial and temporal features of gestures, we propose using a three-input CNN in conjunction with an LSTM to process the input data and perform feature extraction. First, the three-input CNN is used to extract the spatial features corresponding to the low-level representation of the images. Each representation is processed separately in a CNN branch. To fully represent the gesture, the output features of the three CNN branches are merged. The concatenated features are reshaped and provided as input to the LSTM for temporal feature extraction. The LSTM captures and memorizes how the features extracted by the CNN layers change over time. The output of the LSTM is put into vector form and fed into the multiclass SVM. The SVM acts as the final classifier of the architecture and gives the prediction result (Fig. 3).

The noteworthy characteristics of the three-input CNN-LSTM-SVM are:

- Increasing the amount of input data to prevent overfitting and improve the generalization performance of the model.
- Adopting a more efficient feature extractor.
- Using both spatial and temporal information to describe a gesture.
- Providing the final classifier with more information to make a decision.

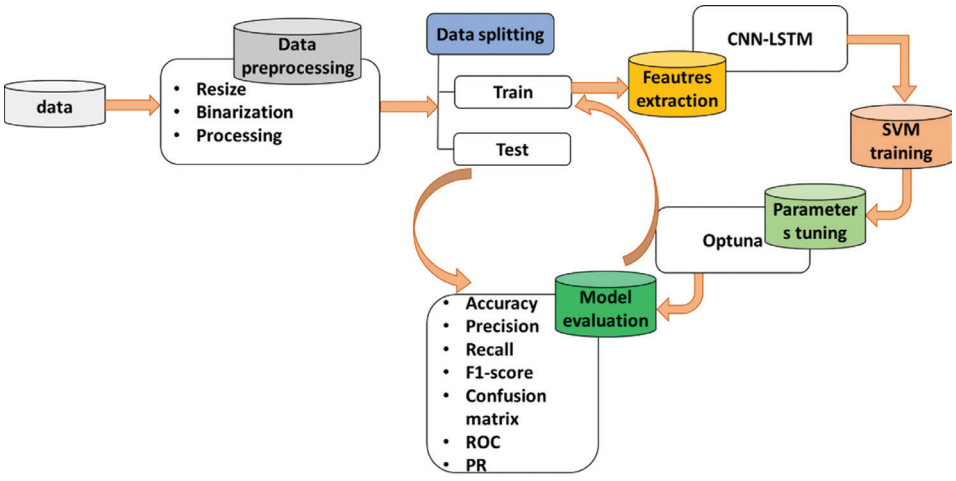

Fig. 3. The overall structure of the proposed hand gesture recognition system

3.3.1. Learning spatiotemporal features

The combination of both spatial and temporal features is a requirement for dynamic gesture classification. To achieve this, two models are combined: the three-input CNN and the LSTM.

The three-input CNN is used for spatial feature extraction. It consists of three branches with similar layer configurations. Each branch consists of an input layer of size 75x75 and three convolutional layers with 16, 32, and 64 filters of size 4x4, respectively, which reduces the number of the training parameters. Additionally, 2x2 strides

are used to further reduce the computational cost. Each branch takes a different image representation as input. The images are processed separately by performing multiple convolution operations to extract spatial features. The convolution operation is expressed as follows:

$$F(i,j) = (I * K)(i,j) = \sum\sum I(i+m, j+n)K(m,n) \quad (1)$$

where I is the input image, i and j are the height and width respectively, K is the 2D convolutional filter of size $m\ x\ n$, and F is the output 2D feature map. To increase nonlinearity in feature maps, a Maxout layer is inserted. It is mathematically expressed as follows:

$$f(x) = max_{j\epsilon[1,k]} Zij \quad (2)$$

$$Zij = x^T W_{ij} + b_{ij}.$$

with x is the input variables, W the weight, and b the bias.

According to the article published by Goodfellow et al. [38], the Maxout activation function has demonstrated its effectiveness for training with Dropout, as well as its robustness for image classification [39]. Each feature map is dimensionally reduced using a Max-pooling layer with a pool size of 2x2 to preserve the most relevant features identified and avoid unnecessary computations. Finally, a Dropout layer with a value of 25% is inserted. The three CNN branches operate in parallel, and their outputs are combined by late fusion for further processing by the LSTM.

The LSTM layer is composed of 150 units. The structure of an LSTM unit consists of input, output, and forget gates that control the learning process, as shown in Fig. 4. These gates are adjusted using the activation sigmoid function. To avoid overfitting in the recurrent layer, the recurrent dropout is set to 0.2.

3.3.2. Multi-class SVM for gesture classification

The spatiotemporal features automatically generated by the CNN-LSTM are fed into the SVM module for training and testing on the hand gesture dataset (Fig. 5). Since the dataset we are using consists of M = 12 classes, we have implemented a multi-class SVM algorithm based on the combination of a set of M binary SVMs. We decomposed the multiclass problem into 64 filters of size 4x4, respectively, which reduces the number of the training parameters. Additionally, 2x2 strides are used to further reduce the computational cost. Each branch takes a different image representation as several bi-class problems and adopted "*the one against all*" strategy, where each binary classifier is trained on the samples of a selected class against all other classes.

This means that the samples on which a classifier is trained are labeled as positive, and all the rest are labeled as negative. In the evaluation phase, a test sample is labeled as belonging to a class according to the maximum score among the 12 classifiers.

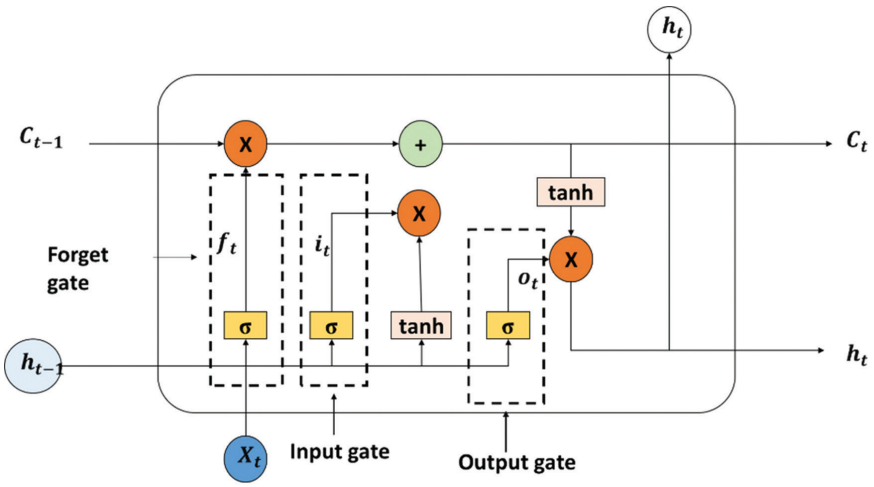

Fig. 4. LSTM structure

3.3.3. Evaluation method

To evaluate the performance of our proposed model, we use the following metrics: accuracy, precision, recall, and F1-score. These metrics are calculated based on the number of true positives (T_p), true negatives (T_N), false positives (F_p), and false negatives (F_N) using the following equations:

$$Accuracy = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \quad (3)$$

$$Precision = \frac{T_P}{T_P + F_P} \quad (4)$$

$$Recall = \frac{T_P}{T_P + F_N} \quad (5)$$

$$F1\text{-}score = \frac{2T_P}{2T_P + F_P + F_N} \quad (6)$$

3.3.4. Implementation details

The three-input CNN-LSTM-SVM model is implemented in Python using the Keras framework with Tensorflow on a machine running an environment with an Intel (R) Core (TM) i5 2.40 GHz CPU, 16GBs of RAM, 1TB of hard disk, and Windows 10. The dataset samples are

randomly split into 80% for training and 20% for testing using the train_test_split function included in the sklearn python package. The random seed parameter is also used to ensure that the test samples are the same in each performed experiment. During the training process of the three-input CNN-LSTM, the Adam optimizer is used with a learning rate set to 0.001, a batch size of 16, and 25 epochs. The input labels are provided as integers rather than vectors to save time in memory as well as computations. Therefore, we use the sparse categorical cross-entropy loss function. The resulting spatiotemporal feature vectors from the three-input CNN-LSTM are fed to the multi-class SVM classifier for training and testing. The learning process of the multi-class SVM classifier is performed using the Optuna framework [40]. During the learning process, the hyperparameters tuning, including the penalty coefficient C, kernel function, slack variable (degree), and gamma parameter, is achieved based on several repeated trials using k cross-validation with k = 5. Optuna is an open-source optimization software available as a library in Python that is easy to implement and offers an integrated dashboard to visualize optimization histories and results. The source code of our model is available on GitHub [41].

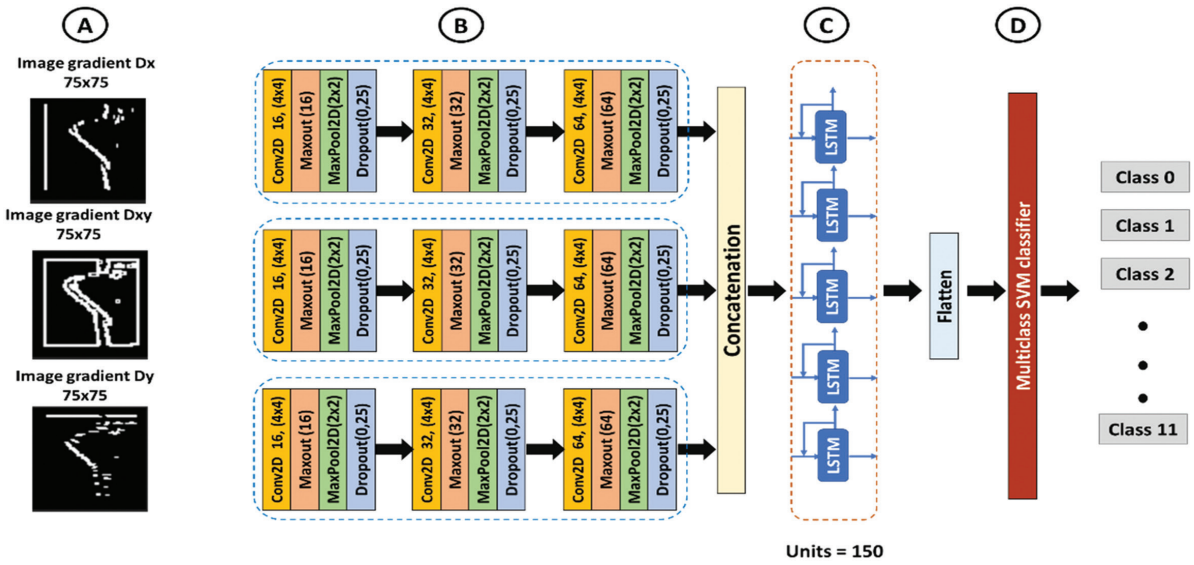

Fig. 5. Three-input CNN-LSTM-SVM model:(A) Input layer, (B) CNN layers for spatial features extraction, (C) LSTM layer for learning temporal features, (D) Classification/output layer.

4. EXPERIMENTAL RESULTS

4.1. TRAINING PROCESS

The training process is divided into three main phases:

- The CNN-LSTM is trained with a SoftMax classifier for the extraction of spatiotemporal features.
- The SoftMax is replaced by the Multiclass SVM, which is fed with the feature vectors, trained, and optimized for 100 trials.
- The Multiclass SVM is then trained for a second time with the optimal hyperparameters.

The finetuned Multiclass SVM achieved a training accuracy of 99.62% (Fig. 6). The influence of different hyperparameters values on the model's performance is presented in Fig. 7.

4.2. EVALUATION PROCESS

The finetuned three-input CNN-LSTM-SVM model is evaluated on the test set. The confusion matrix and the classification report obtained from the test data are depicted below (Fig. 8). Additionally, Receiver Operating Characteristics (ROC) and Precision- Recall (PR) curves are plotted to compare the overall performance (Fig. 9).

4.3. COMPARISON

4.3.1. Verification of data processing

To demonstrate the effectiveness of the proposed data processing approach, the first experiment is divided into two main parts. In the first part, the three-input CNN-LSTM-SVM model is trained using raw images (original images without processing), where the same sample is simultaneously provided to all three CNN branches. In the second part, the model is trained using extended images, where each CNN branch is fed with a different representation of low-level features. The results are shown in Table 1.

Table 1. Comparative classification performance on Raw/Processed images.

Metrics	Raw images	Processed images
Train accuracy	98.34%	99.62%
Test accuracy	92.49%	98.27%
Precision	92.77%	98.30%
Recall	92.40%	98.29%
F1-score	92.44%	98.27%

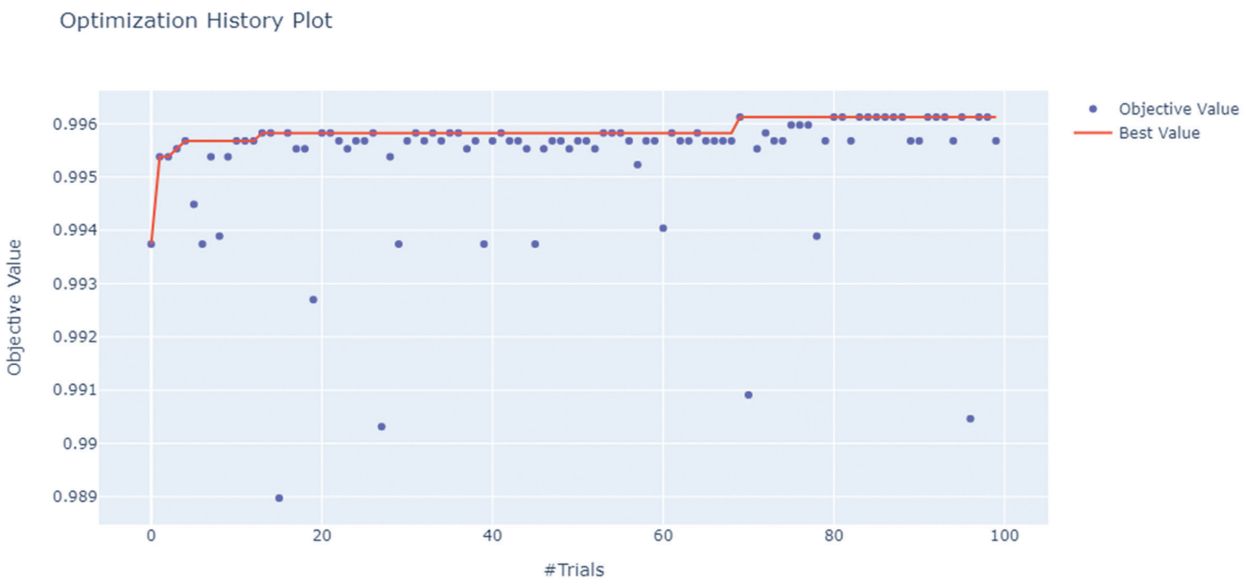

Fig. 6. Optimization history.

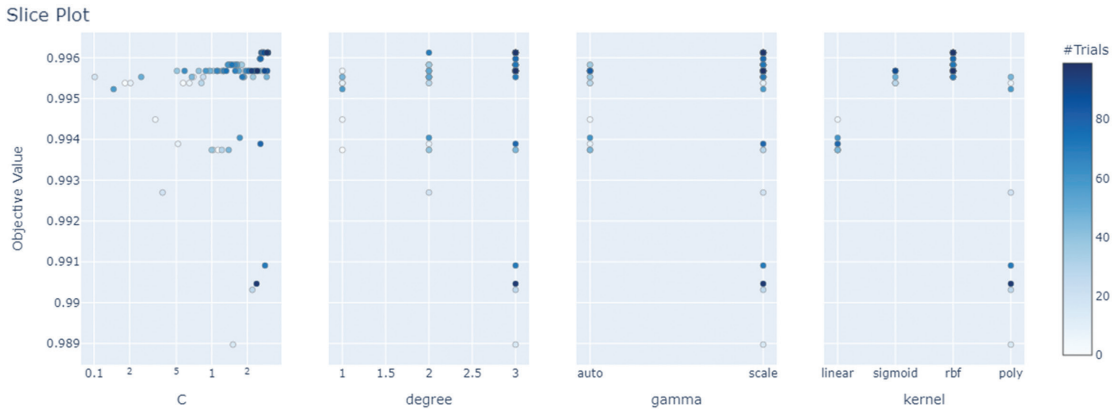

Fig. 7. Slice plot.

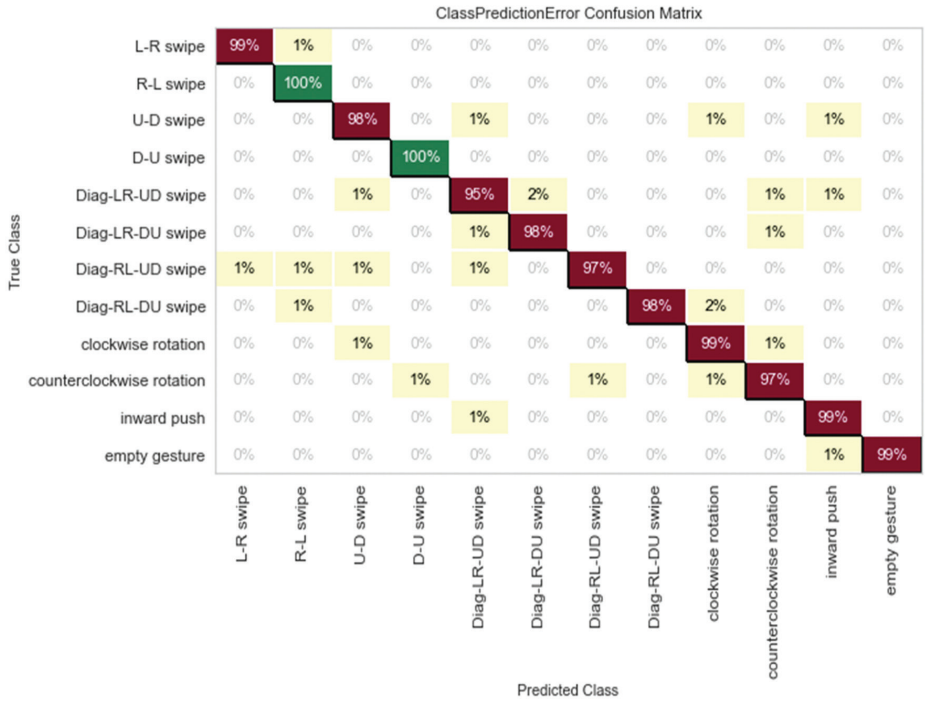

Fig. 8. (a) Confusion matrix

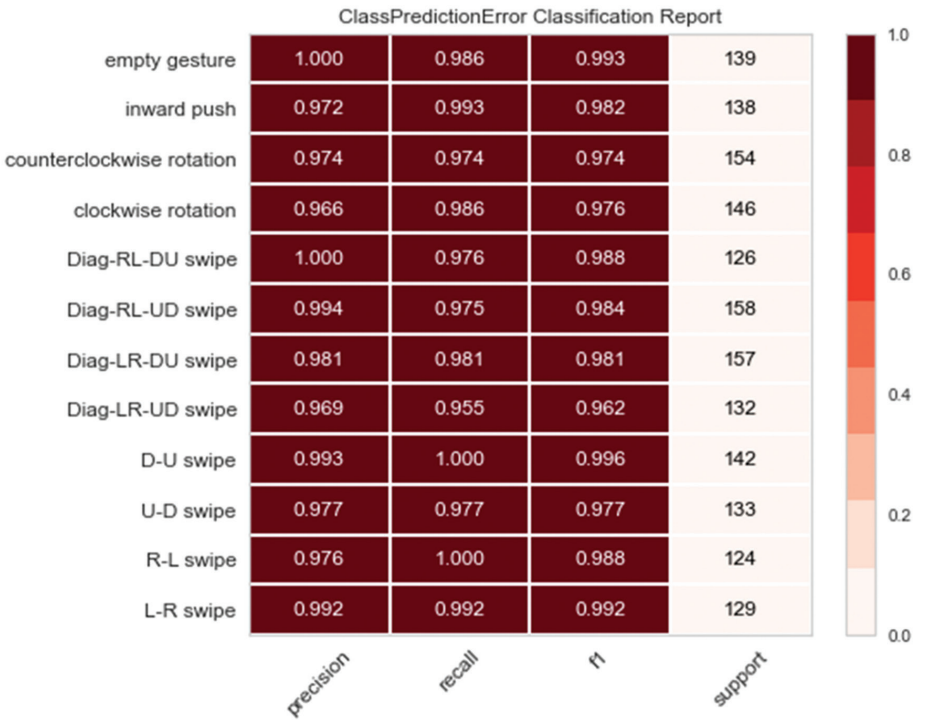

Fig. 8. (b) Classification report

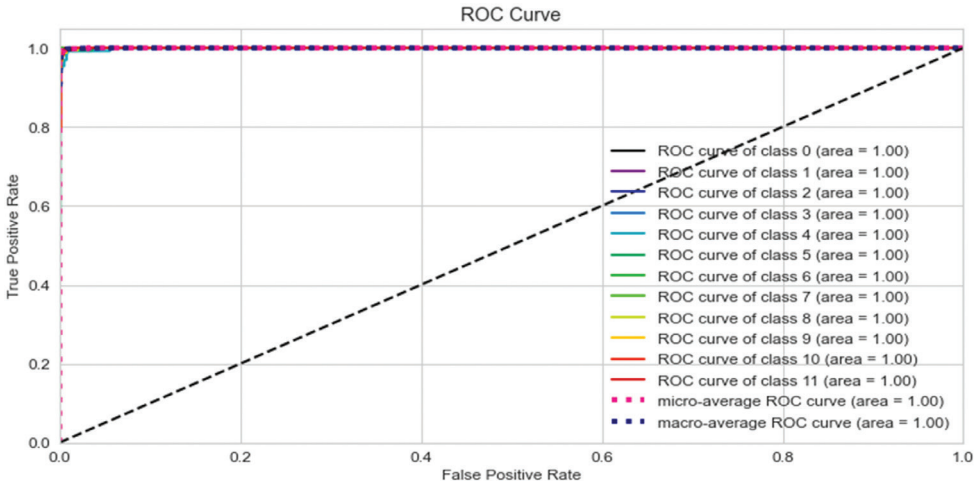

Fig. 9. (a) ROC curve

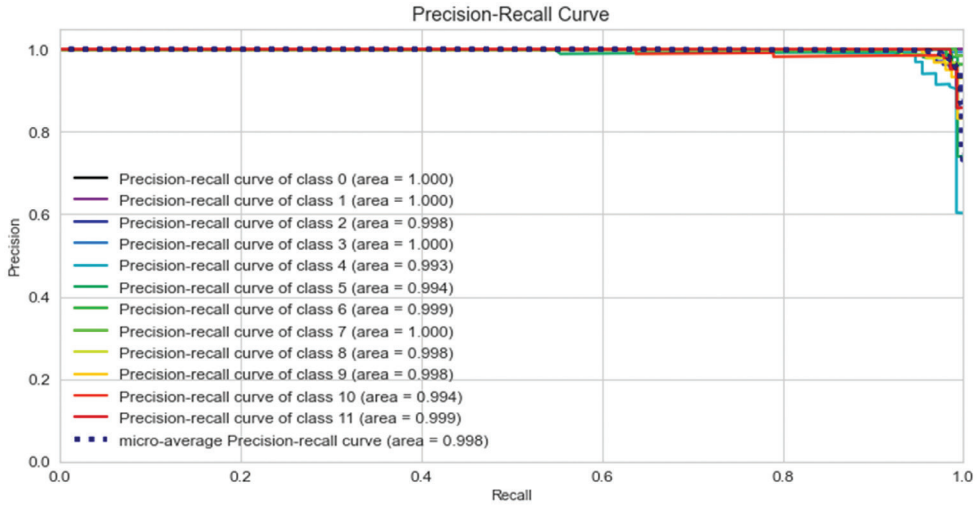

Fig. 9. (b) PR curve

4.3.2. Verification of the model structure

The second experiment aims to demonstrate the superiority of using multiple branches for parallel data processing over a single branch for sequential data processing when using different feature representations. We fed a single-input CNN-LSTM-SVM with the extended images and compared its performance to the results obtained from the three-input CNN-LSTM-SVM experiment mentioned in subsection 4.3.1. Note that the single-input CNN-LSTM-SVM used in this experiment consists of the same layer and parameter configuration as a single CNN branch from the three-input CNN-LSTM-SVM. The results are presented in Table 2.

Table 2. Comparative classification performance of single/three-input CNN-LSTM-SVM.

Metrics	single-input CNN-LSTM-SVM	three-input CNN-LSTM-SVM
Train accuracy	96.30%	99.62%
Test accuracy	93.09%	98.27%
Precision	93.24%	98.30%
Recall	93.10%	98.29%
F1-score	93.44%	98.27%

4.3.3 Comparison to state-of-the-art approaches

The third experiment aims to compare the classification performance of the three-input CNN-LSTM-SVM with other models, including three-input CNN-SoftMax and three-input CNN-LSTM-SoftMax models. This experiment is performed to demonstrate the impact of using spatial features only versus using spatiotemporal features on the recognition rate. Moreover, we aim to find the optimal classifier for the model by comparing SoftMax and SVM. Furthermore, our proposed model results can be directly compared with those of Ahmed et al. [32] and Noori et al. [34], as they also used the same dataset. The results are shown in Table 3.

Table 3. Comparative classification performance of three-input CNN-LSTM-SVM with state of art methods

Reference	Model	Accuracy	N° of parameters
Ahmed et al. [32]	CNN	94%	-
	Three input-CNN-SoftMax	95.41%	126300
Noori et al. [34]	LSTM	97%	847055
	Three input-CNN-LSTM-SoftMax	97.20%	331596
Our approach	Three input-CNN-LSTM-SVM	98.27%	332578

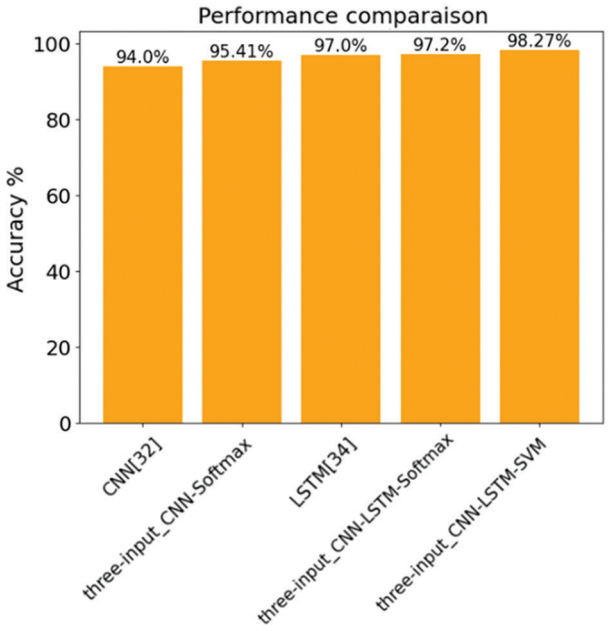

Fig. 10. Comparison of proposed three-input CNN-LSTM-SVM model accuracy with state of art methods.

5. DISCUSSION

This work presents an end-to-end hybrid model for classifying dynamic hand gestures using IR-UWB data. We used a three-input CNN-LSTM for automatically learning spatiotemporal features, combined with a multiclass SVM classifier. The performance of the proposed model was assessed using various evaluations, and the results of the confusion matrix and classification report are depicted in Fig. 8. They show that the model achieved an accuracy of 98.27% and identified half of the classes with an accuracy above 98%, a precision of 98.30%, a recall of 98.29%, and an F1-score of 98.27%. Moreover, the ROC curve in Fig. 9 indicates that the proposed model based on combined features produced excellent results, with a true positive rate produced for each gesture class.

From Table 1, it is evident that the three-input CNN-LSTM-SVM model is overfitting when duplicating inputs using raw images, as it achieved a training accuracy of 98.34% but a lower test accuracy of 92.42%. This is due to the non-representative dataset, which does not provide enough information for the model to generalize well. To address this issue and prevent the model from overfitting, we used the extended dataset that includes image gradients in the x-direction, y-direction, and both x and y-directions. As shown in Table 1, the three-input CNN-LSTM-SVM model achieves better gesture recognition when different low-level representations are provided as inputs, leading to an increase in accuracy of about 6% compared to using duplicate original images. The model achieved a training and test accuracy of 99.62% and 98.27%, respectively, with

much lower overfitting and better generalization ability. This is because using multiple representations for the same gesture enables the extraction and merging of more features, providing the final classifier with more information to make a decision. These results highlight the importance of using different low-level representations as well as a multiple inputs model in cases where there are insufficient representative features, and indicate that unnecessary duplication of inputs does not significantly improve performance but increases computational complexity.

The superiority of our method is more apparent when we compare the performance of the single-input and three-input models. The results in Table 2 show a significant difference in the model's performance when processing the data sequentially and independently in parallel. The single-input CNN-LSTM-SVM achieved an accuracy of 93.09%, while the three-input CNN-LSTM-SVM achieved an accuracy of 98.27%. Compared to the single-input CNN-LSTM-SVM, our proposed three-input CNN-LSTM-SVM architecture not only leverages the strength of multiple low-level representations to extract complementary features from the same target but also introduces the concept of feature concatenation into the architecture to achieve more holistic representations. The separate processing of each data representation allows the extraction and preservation of its features without altering them with other data representations, enabling the model to learn distinct, discriminative, and complementary features effectively. We can conclude that in limited data conditions, extending raw data to different feature representations and then providing them as inputs should be applied to separate branches to achieve higher accuracy. We also hypothesize that the reason for the low performance of the single-input CNN-LSTM-SVM model is due to dissimilar features, where the same gesture is represented three times differently, leading to confusion for the model and making it error-prone. We can reasonably conclude from this experiment that through the effective concatenation of multiple feature information, the prediction is made based on the full use of target features, which improves the recognition accuracy.

On the other hand, as shown in Table 3, the proposed three-input CNN-LSTM-SVM outperforms the most recent research works (Fig. 10) [32,34]. Compared to the CNN proposed by Ahmed et al. [32], the three-input CNN-SoftMax achieved an average increase in accuracy of 1.41%. This result suggests that the three-input CNN-SoftMax can capture more spatial context information for classification by learning the details when a large number of training samples are provided. Although data expansion contributes to the model, the accuracy is still limited by the lack of additional information. While the CNN structure can learn higher-level features, it ignores the temporal dependencies on the features, meaning that the inputs and outputs are independent, leading to limited recognition performance.

We can observe from Table 3 that adding LSTM units can boost classification performance. The combination of three-input CNN-LSTM-SoftMax showed its effectiveness by achieving an accuracy of 97.20%, outperforming the CNN proposed by Ahmed et al. [32] by 3.20% and the three-input CNN-SoftMax by 1.79%. This enhancement refers to convolution, concatenation operations, and the sophisticated structure of LSTM, that maintain the spatial and temporal relationships. The LSTM is cascaded to learn and integrate temporal features, which can provide additional information and improve classification performance. The LSTM helps capture and memorize how the features extracted by the CNN layers change over time. Combining the strengths of CNN and LSTM provides the benefits of both spatial and temporal learning, which is very effective in improving the recognition rate of dynamic hand gestures. Moreover, the three-input CNN-LSTM-SoftMax model provided comparable performance to previous work by Noori et al. [34]. However, the three-input CNN-LSTM-SoftMax model achieved 97.20% accuracy and had fewer trainable parameters (331,596) than the model proposed by Noori et al. [34], which was maintained with 847,055 parameters and achieved 97% accuracy. To select the optimal classifier for our model, the three-input CNN-LSTM was trained with a SoftMax layer and an SVM classifier. The results reported in Table 3 show a 1.07% increase in accuracy using the SVM as the final classifier. This gain is mainly due to the use of the various optimal hyperparameters selected using the Optuna framework. Therefore, the generalization ability of SVM is superior to that of SoftMax.

6. CONCLUSION

A major concern when training deep learning models is the requirement for a large amount of data to achieve sufficient robustness. Otherwise, with limited data, models are prone to overfitting. To enrich the training and testing samples in radar-based HGR, this paper proposes a simple and efficient method to extend radar spectrograms. Using different low-level feature representations as input, processed on separate branches, helps the model learn and merge more information about the target, resulting in increased accuracy. Combining CNN and LSTM layers to take into account both spatial and temporal features improve recognition accuracy. Finally, switching from SoftMax to SVM appears to be beneficial for generalization ability.

We believe that the number of samples, as well as their representation in the dataset, are critical factors in developing a robust model that provides high classification predictions. Future work in this research aims to introduce more data processing techniques to generate additional samples to enhance the model's performance in terms of similar evaluation parameters. Additionally, we plan to further reduce the model's complexity by modifying the layer configuration.

7. REFERENCES

[1] L. Guo, Z. Lu, L. Yao, "Human-Machine Interaction Sensing Technology Based on Hand Gesture Recognition: A Review", IEEE Transactions on Human-Machine Systems, Vol. 51, No. 4, 2021, pp. 300-309.

[2] B. Van Amsterdam, M. J. Clarkson, D. Stoyanov, "Gesture Recognition in Robotic Surgery: A Review", IEEE Transactions on Biomedical Engineering, Vol. 68, No. 6, 2021, pp. 2021-2035.

[3] S. Wu, Z. Li, S. Li, Q. Liu, W. Wu, "An overview of gesture recognition", Proceedings of the International Conference on Computer Application and Information Security, Wuhan, China, 23-24 December 2022, pp. 600-606.

[4] M. Pan, Y. Tang, H. Li, "State-of-the-Art in Data Gloves: A Review of Hardware, Algorithms, and Applications", IEEE Transactions on Instrumentation and Measurement, Vol. 72, 2023, pp. 1-15.

[5] B. K. Chakraborty, D. Sarma, M. K. Bhuyan, K. F. MacDorman, "Review of constraints on vision-based gesture recognition for human-computer interaction", Institute of Engineering and Technology Computer Vision, Vol. 12, No. 1, 2018, pp. 3-15.

[6] S. Ahmed, K. D. Kallu, S. H. Cho, "Hand gestures recognition using radar sensors for human computer-interaction: A review", Remote Sensing, Vol. 13, No. 3, 2021, p. 527.

[7] S. Ahmed, F. Khan, A. Ghaffar, F. Hussain, S. H. Cho, "Finger-counting-based gesture recognition within cars using impulse radar with convolutional neural network", Sensors, Vol. 19, No. 6, 2019, pp. 1429.

[8] S. K. Leem, F. Khan, S. H. Cho, "Detecting Mid-Air Gestures for Digit Writing with Radio Sensors and a CNN", IEEE Transactions on Instrumentation and Measurement, Vol. 69, No. 4, 2020, pp. 1066-1081.

[9] N. Hendy, H. M. Fayek, A. Al-Hourani, "Deep Learning Approaches for Air-Writing Using Single UWB Radar", IEEE Sensors Journal, Vol. 22, No. 12, 2022, pp. 11989-12001.

[10] H. Hameed, M. Usman, M. Z. Khan, A. Hussain, H. Abbas, M.A. Imran, Q. H Abbasi, "Privacy-Preserving British Sign Language Recognition Using Deep Learning", Proceedings of the 44th International Conference of the IEEE Engineering in Medicine & Biology Society, Glasgow, Scotland, UK, 11-15 July 2022, pp. 4316-4319.

[11] Y. Yang, J. Li, B. Li, Y. Zhang, "MDHandNet: a lightweight deep neural network for hand gesture/sign language recognition based on micro-doppler images", World Wide Web, Vol. 25, No. 5, 2022, p. 1951-1969.

[12] D. Sarma, M. K. Bhuyan, "Methods, Databases and Recent Advancement of Vision-Based Hand Gesture Recognition for HCI Systems: A Review", SN Computer Science, Vol. 2, No. 6, 2021, p. 436.

[13] Y. Shi, Y. Li, X. Fu, K. Miao, Q. Miao, "Review of dynamic gesture recognition", Virtual Reality & Intelligent Hardware, Vol. 3, No. 3, 2021, pp. 183-206.

[14] J. Park, S. H. Cho, "IR-UWB radar sensor for human gesture recognition by using machine learning", Proceedings of the 18th International Conference on High Performance Computing and Communications, 14th International Conference on Smart City, 2nd International Conference on Data Science and Systems, Sydney, NSW, Australia, 12-14 December 2016, pp. 1246-1249.

[15] K. Faheem, L. Seong Kyu, S. H. Cho, "Algorithm for fingers counting gestures using IR- UWB radar sensor", Proceedings of the International IEEE Sensors Applications Symposium, Seoul, Korea, 12-14 March 2018, pp.144-146.

[16] S. Y. Kim, H. G. Han, J. W. Kim, S. Lee, T. W. Kim, "A hand gesture recognition sensor using reflected impulses", IEEE Sensors Journal, Vol. 17, No. 10, 2017, pp. 2975-2976.

[17] B. Li, J. Yang, Y. Yang, C. Li, Y. Zhang, "Sign Language/Gesture Recognition Based on Cumulative Distribution Density Features Using UWB Radar", IEEE Transactions on Instrumentation and Measurement, Vol. 70, 2021, pp. 1-13.

[18] F. Khan, S. Leem, S. H. Cho, "Hand-based gesture recognition for vehicular applications using IR-UWB radar", Sensors, Vol. 17, No. 4, 2017, p. 883.

[19] A. Ghaffar, F. Khan, S. H. Cho, "Hand Pointing Gestures Based Digital Menu Board Implementation Using IR-UWB Transceivers", IEEE Access, Vol. 7, 2019, pp. 58148-58157.

[20] L. Yao, W. Xin, S. Baodai, M. Zhu, "Hand Gesture Recognition Using IR-UWB Radar with ShuffleNet V2", Proceedings of the 5th International Conference on Control Engineering and Artificial Intelligence, Sanya, China, 14-16 January 2021, pp. 126-131.

[21] A. Bhavana, K. S. R. Kumar, M. D. Praveen, "Deep Neural Network based Sign Language Detection", Proceedings of the 6 th International Conference on Electronics, Communication and Aerospace Technology, Coimbatore, India, 1-3 December 2022, pp. 1474-1479.

[22] J. J. Ojeda-Castelo, M. D. L. M. Capobianco-Uriarte, J. A. Piedra-Fernandez, R. Ayala, "A Survey on Intelligent Gesture Recognition Techniques", IEEE Access, Vol. 10, 2022, pp. 87135-87156.

[23] Z. Li, W. Yang, S. Peng, F. Liu, "A Survey of Convolutional Neural Networks: Analysis, Applications and Prospects", IEEE Transactions on Neural Networks and Learning Systems, Vol. 33, No. 12, 2021, pp. 6999-7019.

[24] S. Ahmed, S. H. Cho, "Hand gesture recognition using an IR-UWB radar with an inception module-based classifier", Sensors, Vol. 20, No. 2, 2020, p. 564.

[25] G. Park, V. K. Chandrasegar, J. Park, J. Koh, "Increasing Accuracy of Hand Gesture Recognition using Convolutional Neural Network", Proceedings of the International Conference on Artificial Intelligence in Information and Communication, Jeju Island, Korea, 21-24 February 2022, pp. 251-255.

[26] G. Park, V. K. Chandrasegar, J. Koh, "Accuracy Enhancement of Hand Gesture Recognition using CNN," IEEE Access, Vol. 11, 2023, pp. 26496-26501.

[27] Y. Yu, X. Si, C. Hu, J. Zhang, "A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures", Neural Computation, Vol. 31, No. 7, 2021, pp. 1235-1270.

[28] H. Liu, Z. Liu, "A Multi-Modal Dynamic Hand Gesture Recognition Based on Radar-Vision Fusion", IEEE Transactions on Instrumentation and Measurement, Vol. 72, 2023, pp. 1-15.

[29] S. Skaria, A. Al-Hourani, Da Huang, "Radar-Thermal Sensor Fusion Methods for Deep Learning Hand Gesture Recognition", Proceedings of the International Conference of IEEE Sensors, Sydney, Australia, 31 October - 3 November 2021, pp. 1-4.

[30] L. O. Fhager, S. Heunisch, H. Dahlberg, A. Evertsson, L. E. Wernersson, "Pulsed Millimeter Wave Radar for Hand Gesture Sensing and Classification", IEEE Sensors Letters, Vol. 3, No. 12, 2019, pp. 1-4.

[31] S. Z. Gurbuz, M. G. Amin, "Radar-based human-motion recognition with deep learning: Promising applications for indoor monitoring", IEEE Signal Processing Magazine, Vol. 36, No. 4, 2019, pp. 16-28.

[32] S. Ahmed, D. Wang, J. Park, S. H. Cho, "UWB-gestures, a public dataset of dynamic hand gestures acquired using impulse radar sensors", Scientific Data, Vol. 8, No. 1, 2021, pp. 1-9.

[33] F. Khan, S. K. Leem, S. H. Cho, "In-Air Continuous Writing Using UWB Impulse Radar Sensors", IEEE Access, Vol. 8, 2020, pp. 99302-99311.

[34] F. M. Noori, M. Z. Uddin, J. Torresen, "Ultra-Wideband Radar-Based Activity Recognition Using Deep Learning", IEEE Access, Vol. 9, 2021, pp. 138132-138143.

[35] J. Park, J. Jang, G. Lee, H. Koh, C. Kim, T. W. Kim, "A Time Domain Artificial Intelligence Radar System Using 33-GHz Direct Sampling for Hand Gesture Recognition", IEEE Journal of Solid-State Circuit, Vol. 55, No. 4, 2020, pp. 879-888.

[36] S. Skaria, A. Al-Hourani, R. J. Evans, "Deep-learning methods for hand-gesture recognition using ultra-wideband radar", IEEE Access, Vol. 8, 2020, pp. 203580-203590.

[37] S. Skaria, D. Huang, A. Al-Hourani, R. J. Evans, M. Lech, "Deep-Learning for Hand-Gesture Recognition with Simultaneous Thermal and Radar Sensors", Proceedings of the International Conference of IEEE Sensors Conference, Rotterdam, Netherlands, 25-28 October 2020, pp. 1-4.

[38] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, Y. Bengio, "Maxout networks", Proceedings of the International Conference on Machine Learning Research, Atlanta, Georgia, USA, 17-19 June 2013, pp. 1319-1327.

[39] R. Rajeswari, M. Prabhakar, G. Padmapriya, B. S. Kumar, "Blood vessel detection using enhanced DeepJoint fuzzy clustering algorithm with deep Maxout network for glaucoma detection", Concurrency and Computation: Practice and Experience, Vol. 35, No. 6, 2023, p. 1.

[40] T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, "Optuna: A Next-generation Hyperparameter Optimization Framework", Proceedings of the 25th International Conference on Knowledge discovery & Data mining, Anchorage, AK, USA, 4-8 August 2019, pp. 2623-2631.

[41] GitHub, three_input_CNN_LSTM_SVM, https://github.com/souhila1998/HGR_CNN-LSTM-SVM (accessed: 2023)