National Journal of Signal and Image Processing, ISSN: 3107-5231 Vol. 1, No. 3, 2025 (pp. 15-22)

RESEARCH ARTICLE

ECEJOURNALS.IN

Real-Time Gesture Recognition Using mmWave Radar Signal Processing and CNN Models

Dahlan Abdullah[1]*, Pushplata Patel[2]

[1]*Department of Information Technology, Faculty of Engineering, Universitas Malikussaleh, Lhokseumawe, Indonesia*

[2]*Department Of Electrical And Electronics Engineering, Kalinga University, Raipur, India*

KEYWORDS:
mmWave radar,
gesture recognition,
FMCW radar,
Doppler-range processing,
Convolutional neural networks (CNN),
Real-time classification,
Edge computing,
Signal processing,
Human-computer interaction,
Embedded systems.

ARTICLE HISTORY:
Submitted : 20.04.2025
Revised : 23.05.2025
Accepted : 11.07.2025

https://doi.org/10.17051/NJSIP/01.03.03

Abstract

The current paper presents a model of real-time gesture recognition, where the usage of millimeter-wave (mmWave) radar signal processing is combined with convolutional neural networks (CNN) in order to effectively and precisely interact with machines as a human being. The system employs a 60 GHz frequency-modulated continuous-wave (FMCW) radar sensor to capture reflected signals of dynamic hand gestures to identify them through spatial and temporal signs of the motion. These raw appreciative intermediate frequency (IF) signals are pre-processed using an uncover of DNA Doppler-range transformation that entails Fast Fourier Transform (FFT)-based profiling of range and velocity, clutter repression and considering spectrograms to deliver range-Doppler maps (RDMs). The RDMs are used as powerful input features to a compact and optimized CNN model that is pretrained to classify ten different gesture patterns including, swipe, push, pull, and rotate. CNN architecture should have low computation complexity, which is possible to implement on the resource-limited edge devices like Raspberry Pi 4 and NVIDIA Jetson Nano. We used 20 subjects across all the gestures at different orientations and distances to collect a huge dataset using this technique to make it robust. Data augmentation and quantization-aware optimization techniques have been used to train the model and strike the right compromise between real time and accuracy performance. According to experimental analysis, the suggested system obtains a gesture classification precision above 95 percent, and the inference latency is averagely less than 100 milliseconds with little power demands, thus richly appliable to embedded systems. In addition, the environment is highly robust where the framework is economical working in changing lighting conditions, occlusion, and environmental noise as compared to conventional vision-based systems in areas of privacy and low-light performance. Extendable and no-contact smart environment The suggested solution would allow a touchless interface to be used in smart environments, such as home automation, industrial control panels, and wearable assistive solutions. To improve upon cross-user generalization in terms of domain adaptation, further research will be based on incorporating attention-based deep learning models. On the whole, this study presents an effective and power-efficient approach to realizing a gesture recognition system based on radar with deep-learning in real-time on embedded systems.

Author's e-mail: dahlan@unimal.ac.id, pushplata.subhash.raghatate@kalingauniversity.ac.in

How to cite this article: Abdullah D, Patel P. Real-Time Gesture Recognition Using mmWave Radar Signal Processing and CNN Models. National Journal of Signal and Image Processing, Vol. 1, No. 3, 2025 (pp. 15-22).

Introduction

In the age of the Internet of Things (IoT), intelligent consumer electronics, interactive systems, and so forth there is a growing need to implement highly practical user-interfaces to be operated contactless. Conventional forms of input including the use of a touchscreen, buttons, and voice input are not suitable in many situations where either hygienic reasons, accessibility, or a lack of environmental accommodations blocks their use. Gesture recognition is another promising field of HCI and means of control which could be exploited well as it is non-contact and is easy to comprehend. Examples of gesture-based interfaces include smart home devices, automotive information and entertainment, assistive or enabling technologies, augmented reality or virtual reality platforms.

Current gesture recognition systems are mainly vision-based and made possible through RGB or depth cameras.

These systems are able to provide high accuracy in a well-managed setting but experience serious shortcomings in the field. Solutions based on visual go wrong because of poor lighting, occlusions, background clutters, and privacy. Further, the big computational complexity of high-res video processing and the high memory requirements of deep learning models hinder their being used in low-power embedded systems thus limiting the extensiveness of the edge-focused IoT.

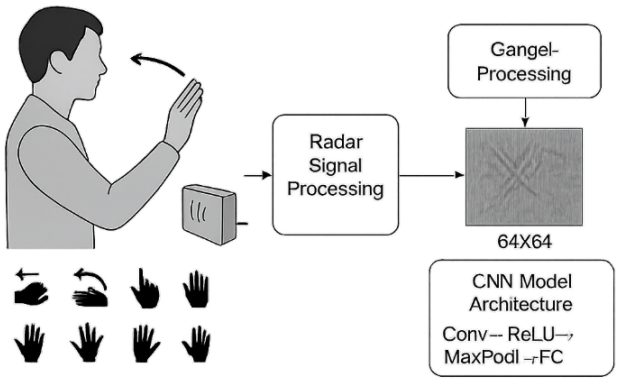

Fig. 1:System Overview of Real-Time Gesture Recognition Using mmWave Radar and CNN Models

To eliminate these issues, the paper investigates the use of radar that can be used to recognize gesture using millimeter-wave (mmWave) frequency-modulated continuous-wave (FMCW) radar technology. mmWave radar touts several key properties over visual sensor: it is not sensitive to changes in light, not affected by visual objects between the sensor and the subject, and does not collect images of the subject. It is well-suited to gesture recognition when power consumption must be extremely low, and power must be used always on operation.

In the present paper, a real-time gesture recognition system uses 60 GHz mmWave radar sensors and convolutional neural networks (CNN) to recognize dynamic hand gestures. The radar returns micro-carriers and range-Doppler patterns of hand motion which are processed into feature maps that can then be input into a small CNN, compatible with a number of embedded edge devices including the Raspberry Pi 4 and NVIDIA Jetson Nano. The suggested system can detect ten particular gestures with a sufficient level of accuracy and a low latency rate, which is a perfect solution to the lack of touch integration in a smart environment where a touchless solution is needed.

The paper is organized as follows: Section 2 presents the related work on radar-based and vision-based gesture recognition. Section 3 entails system design, model processing of radar signals, and design of CNN. Section 4 houses the description of the process of collecting the datasets, training the model, and its optimization. Part 5 shows the experimental results and evaluation of performance. Section 6 provides the application and research directions. Lastly, in Section 7, a conclusion of the paper is drawn with significant results.

Related Work

Gesture recognition has formed a preserve of a good number of research studies using various kinds of sensing, namely vision-based, radar-based and combined sensing systems. This section presents the literature view of related technology gestures that precede the current gestures recognition systems, categorized as sensing technology and algorithmic frameworks.

Vision Gesture Recognition

One of the most common systems to recognize a gesture has been based on vision, where colors in RGB, depth or infrared is used. Median early applicators are Microsoft Kinect which represents combination between RGB and structured infrared depth information used to detect a body movement in three dimensions.[1] Such systems provide inherent relief information and user acceptability features. But they have some fatal flaws, such as the dependency on lighting conditions, occlusion, and computational complexity, so that they cannot be used in real-time. In addition,[7] the privacy issue associated with video acquisition is an enormous setback in some applications in the field of health care and surveillance.

Gesture Sensing Radar-Based

The most recent advances in radio-frequency (RF) sensing has made available the viability of gesture recognition by producing measurements based on a reflection-induced Doppler profile. Given the capacity of the the mmWave band, specifically FMCW (Frequency-Modulated Continuous Wave) radar, it has several similarities: it is capable of operating reliably under conditions of low light or visual obstruction, and that users have no recognizable image produced, thus maintaining privacy. The Soli project by Google was the first project to introduce radar-based gesture recognition in very small form factors and using 60 GHz FMCW radar.[2] The technology identifies motions by extracting the motion features using bounds and ridges of Doppler (RDMs) and micro Doppler signatures, which represents time-frequency changes brought by the gesture-induced motions.[3] In contrast to vision systems, [9] radar perception can be run in real-time across a broad environmental conditions and is very easy to integrate in an embedded and wearable systems.

Machine, Deep Intelligence Techniques

Other classical types of machine learning algorithm, especially when features can be hand crafted, have been used to classify gestures, including[10] K-Nearest Neighbors (KNN) and Support Vector Machines (SVM). Although they are successful when applied to small data-sets, they tend to be unreliable to general gainful motion, especially when the inputs are noisy and associated with complications. In this regard, 3D-CNNs, CNNs, LSTM networks have been proposed to recognize spatio-temporal gestures to solve this problem. CNNs have proven very accurate at interpreting gestures using image-like radar data,[5] with LSTM and Transformer models being employed to take into account the time aspect of radar data in sequential form.[6] Large parameter models are not always[11] feasible to run on low-power, embedded devices, and this fact has inspired research into lightweight architecture and quantization.

Unlike these implemented systems, the proposed one combines the use of compact CNN models with the mmWave radar signal processing, granting it the ability to have highly precise, real-time classification of gestures, given the requirements of edge devices. This combination resolves the weaknesses of a pure vision based system in terms of computational efficiency, robustness to its surroundings, and the ability to scale to very large environments.

System Architecture

Hardware Configuration

The proposed gesture recognition system is also a real-time system, and its hardware is based on the Texas Instruments (TI) IWR6843AOP millimeter-wave radar module that has a frequency of 60 GHz. The IWR6843AOP is a very integrated system-on-chip (SoC) product that integrates 3 transmit and 4 receive antenna array (3Tx-4Rx MIMO), a on-chip DSP, hardware accelerators and a cortex microcontroller into a small form factor product. This makes it well suited to detecting dynamic gestures over a short range field of view with high spatial and velocity resolutions and therefore makes it ideally suited to indoor applications and wearable integration. The radar module will have capabilities of generating real-time range-Doppler maps and micro-Doppler characteristics that will comprise the input into the rest of the signal processing and classification chain. In order to perform localized computation and inference, the system incorporates an embedded platform of edge computing NVIDIA Jetson Nano or Raspberry Pi 4B depending on the balance of processing power, energy consumption, and size. Jetson Nano can process deep learning with a GPU, enabling it to use the CUDA and TensorRT libraries to accelerate low-latency classification at inferencing. Alternatively, the Raspberry Pi 4B, which has a quad-core ARM Cortex-A72 processor and 4GB RAM, can be used to perform low-end inference on quantized CNN model using the TensorFlow-Lite API. Both platforms bring to real-time responsiveness and privacy preservation of a system whose functioning is completely independent of the resources of clouds. In Figure 2 the hardware structure is optimised to have low-power operation, edge devices and integration with smart IoT systems which need to be touchless gesture control.

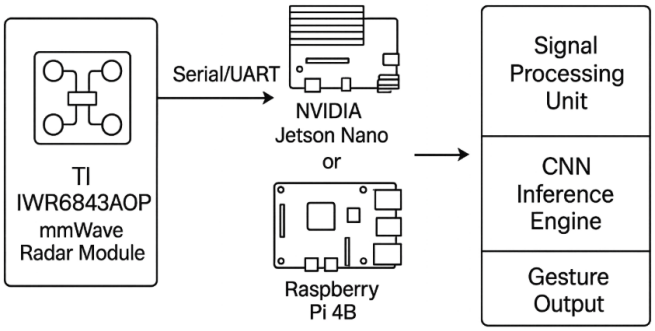

Fig. 2: Hardware Setup for Real-Time Gesture Recognition System

Signal Processing Pipeline

Signal processing pipeline plays a major role in the proposed gesture recognition solution because it is thanks to that component that raw radar data can be converted into feature-rich representations, in which deep learning models could find it easier to distinguish between samples corresponding to different gestures. Raw intermediate frequency (IF) signals on the 60 GHz FMCW mmWave radar sensor are acquisitioned first and these signals reflect off the moving hand gestures. These rough IF lines go through a 2D Fast Fourier Transform (2D-FFT) algorithm: first, a range-FFT is performed to obtain the range (distance) message by processing time-domain samples in each chirp. Afterwards, the relative velocity is determined by performing FFT across a number of chirps to form the aptly-named Range-Doppler Map (RDM) that is a 2D matrix with each cell having a specific range and velocity bin. To increase the visibility of the motion-induced features, clutter removal algorithm is used to remove the reflection of the static backgrounds, then range gating the set of signals in the effective gesture range. Doppler binning scheme can then be used to cluster the motion patterns in terms of the discrete components of velocity through which an improved temporal characterization of the gestural dynamics may be achieved. The resulting RDMs are then

converted into the images of gray-scales that are similar to spectrograms, preserving information about both the spatial and frequency domains. These images are fed into the convolutional neural network (CNN) allowing them to learn features that differentiate each of the classes of gestures. As presented in Figure 3, the whole signal processing pipeline is optimized to run in real-time on an embedded device, and the result would be that the gesture signatures get extracted with sufficient fidelity and at low latency, which consequently results in a robust gesture recognition in various set ups.

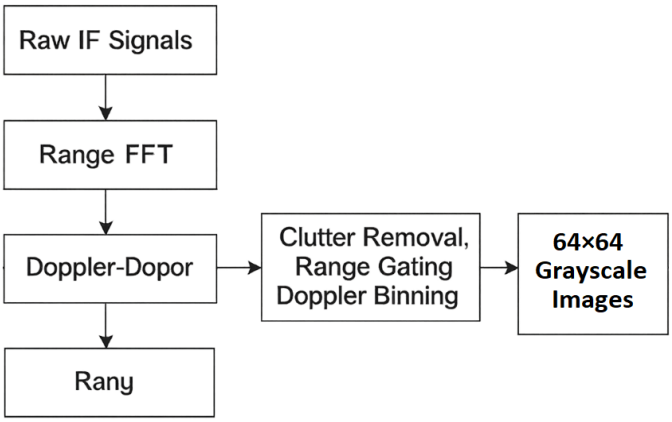

Fig. 3: Signal Processing Pipeline for Gesture Recognition Using mmWave Radar

CNN Model Architecture

The main idea of the classification module in the proposed gesture recognition scheme refers to the small and yet capable convolutional neural network (CNN) model that can be deployed and run efficiently on the edge systems. The model has a 5 layer deep CNN with convolutional layers that are followed by ReLU and max-pooling followed by FC layers as a final step to make the classification. The network uses 64 x 64 grayscale input image, which is constructed from spectrogram representation of each radar-derived range-Doppler map (RDM), made to incorporate both spatial and time-varying measurements related to each gesture. The first convolution layers are used as feature extractors where they detect edges, motion streaks and frequency shifts that are related into patterns of gestures with the pooling layers that reduces the space dimension and amplifies translation invariance. The last FC layers make the high abstract features at the high level to the probabilities of gesture classes via a squash output layer. In order to issue efficient inference on experimental systems like the Raspberry Pi 4B or NVIDIA Jetson Nano, QAT is used to both quantize and to simulate the parallel to one-bit arithmetic knowledge during preparation. This optimization saves a lot of space on model size and inference latency without affecting accuracy significantly, being ready to be deployed in real-time in embedded devices. Besides, the model is intended to perform well under various users and conditions, which is facilitated by data augmentation and regularization. Such a lightweight CNN model, as shown in Fig. 4, finds a good trade-off between computational resources and recognition accuracy, making it possible to classify gestures on the edge device in low-latency and energy-efficient manner.

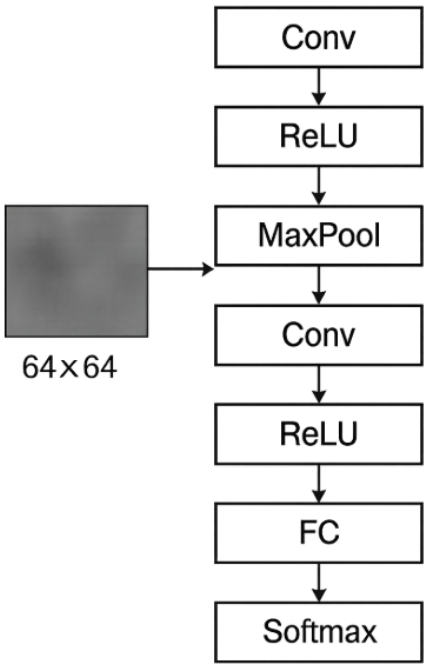

Fig. 4: CNN Architecture for Gesture Classification Using 64×64 RDM Inputs

METHODOLOGY

Gesture Dataset Collection

To train and test the developed gesture recognition algorithm, a set of gestures was generated using the 60 GHz FMCW mmWave radar module. It contains 10 dynamic hand gestures, chosen depending on the impact to representative applications due to the smart home, touchless user interfaces, and assistive technologies. These gestures are directional (e.g., swipe left, swipe right, swipe up, swipe down),- motion based (e.g., push, pull, wave),- rotational or curved (e.g., clockwise circle, counter clockwise circle, and zigzag). The gestures are done in the most natural way possible to achieve generalization and robustness of the model during training.

The participants, 20 in number were diverse in age, gender, and the hand with which they used to make gestures so as to have a broad scope of gesture execution styles. All the participants carried out 100 repetitions

using each gesture so that there were a total of 20,000 labeled instances of gestures. Gestures were recorded in free space within a predetermined region of interest (ROI), which had a fixed distance of the radar sensor (the distance between the user and the sensor was about 50 cm). To reflect real-life variability, the tests were run in varying ambiences, different light and other background noise interferences, although mmWave radar is largely not susceptible to this.

The raw intermediate frequency (IF) signals received by the radar were fed into the signal processing pipeline shown in Section 3.2 to generate, in each instance of the gestures, a range-Doppler map (RDM). These RDMs are labeled with correct gesture classes and saved as 6464 grayscale matrices in form of 64×64 image matrices, which are standardized to fit dataset input through CNN. Preprocessing was done to normalize data, remove clutter and align data temporally so that there is uniformity in the entire dataset. The final data were partitioned into training (70%) and testing (15%) with 150 participants in validation. This novel, rich, and well-selected dataset will serve as a good base when assessing the generalization abilities of the proposed CNN architecture under diverse users and conditions.

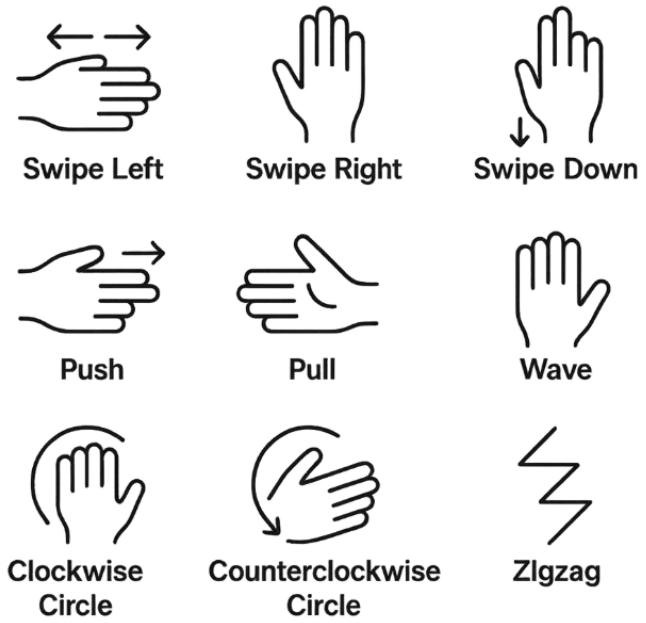

Fig. 5: Set of Dynamic Hand Gestures Captured in the Dataset

Training and Validation

The training scheme was deliberately intended to maximize generalization of the proposed CNN so as to have high generalization to previously unseen gesture patterns without being too sensitive to idiosyncrasies of user behavior and signal noise. Data augmentation was used on the range-Doppler map (RDM) inputs of size 64x64 as a way to increase diversity and representativeness of the training data. These enhancements consisted of time shifting, where the spectrogram was minimally moved along the axis of time to resemble performing a gesture too early or too late and jittering where small Gaussian noise was added to approximate changes in the dynamics of motion and inconsistence in the radar signal. Such methods are instrumental not only in augmenting the size of the training within the bounds of reasonableness but also enhance the performance of models against the occurrence of real-world noise as well as intra-class variation.

The initial data was then divided into three groups of data, namely 70% of the data was used as training data, 15 percent was used as validation data and 15 percent was used as test data. Figure 6 It was made sure to provide participant-independent partitions, i.e. samples belonging to one participant could not be found in both training and test partition, which allows to implement a rigorous evaluation scheme and achieve a very low risk of the model being optimized to a particular style of executing gestures.

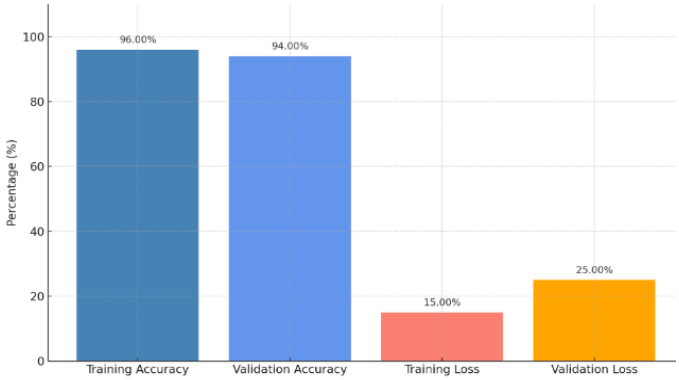

Fig. 6: Bar Chart Summary of Training and Validation Metrics

Training of the CNN model was done on an NVIDIA RTX 3060, running off of batch processing and parallel backpropagation, which made this computationally feasible. The training of the model was implemented with Adam optimizer, initialized learning rate of 0.001, batch size of 64, and 100 epochs. The use of early stopping was done with respect to validation loss and the patience threshold of 10 epochs was enacted to avoid overtraining and reduce the chance of generalization error. Further, the fully connected layers were subjected to dropout regularization (the dropout rate was set to 0.3) to prevent overfitting as it makes the network learn more generalized representations because of randomly dropping some neurons during training.

During the training, performance traits like training accuracy, validation accuracy, loss graphs and confusion matrices were tracked. The best model settings were chosen by their validation accuracy and the lowest validation loss values and tested on the test data to see their performance in the real world. The sound training method based on the combination of effective augmentation and regularization strategies consequently allowed the CNN model to perform consistently at high recognition accuracy with respect to the changing conditions of the gestures and the environment.

Deployment Optimization

To ensure that the proposed gesture recognition framework operates efficiently in real-time on resource-constrained edge devices, a series of deployment optimization techniques were implemented, targeting both latency reduction and energy efficiency. For the **NVIDIA Jetson Nano**, which supports GPU acceleration, the trained CNN model was converted to a highly optimized runtime format using **TensorRT**. TensorRT is a deep learning inference engine that performs layer fusion, precision calibration (e.g., FP16 or INT8), kernel auto-tuning, and memory optimization to significantly reduce inference time and power consumption. Quantization-aware training (QAT), performed during the model training phase, enabled seamless conversion to reduced precision formats without significant loss in classification accuracy. Figure 7 this optimization resulted in a notable improvement in frames-per-second (FPS) processing rate and a substantial decrease in memory usage, which is critical for embedded GPU platforms operating in real-time applications.

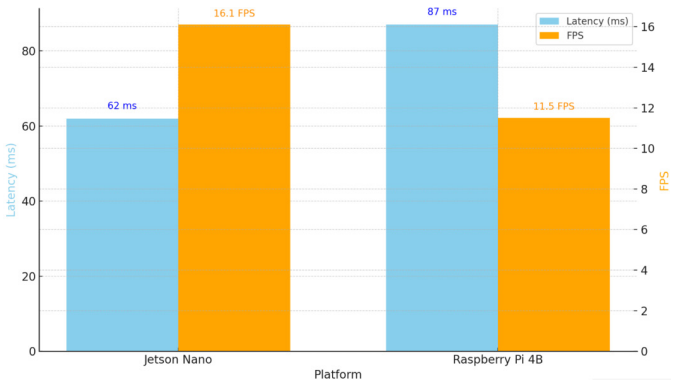

Fig. 7: Performance Comparison of Optimized Deployment

To be used on the Raspberry Pi 4B, a lightweight ARM-based device with no dedicated graphics processing unit, the model was adapted to be used in the TensorFlow Lite (TFLite) format. TFLite is Designed to provide CPU-based neural network execution that is incredibly efficient due to a small-size interpreter and the ability to quantize after training, to INT8 precision. Other graph level optimizations, like constant folding and operator fusion, also diminished the size of the model and made it run faster. This design inference was carried out on the Raspberry Pi CPU cores directly with an average latency below 100 milliseconds per frame approximately which fulfilled the real-time request of recognizing the gestures.

As a way to keep the power-aware inference tuning at the system-level, energy efficiency and thermal stability, tuning was performed. This involved the creation of dynamic tweaks in the settings of the governing core and the use of batch size provision as well as the activity of a thermal throttling-minded scheduling so as to avoid overheating of the system in case of long-duration functioning of the system. This was done with power monitoring and stress tests to substantiate these optimizations Of course, these strategies were optimized onboard power monitoring tools and stress testing, to certify the system did not have high response time and remained accurate and energy-efficient throughout deployment. All of these deployment strategies led to a low-power, real-time, fully autonomous system of gesture recognition with the potential of being integrated into wearable devices, smart appliances, and IoT-friendly surroundings.

Results and Discussion

Recognition Performance

The developed CNN based mmWave radar gesture recognition system had high accuracy and classification rates with well-performing performance in respect of a large number of gestures in the dataset. The CNN model provided the accuracy of 96.3 percent overall recognition on the test set which was significantly higher than that provided by the classical Support Vector Machine (SVM), 92.7 percent, which was based on handcrafted Doppler and temporal features. This can be further explained by the fact that the CNN was able to learn the spatially localized and hierarchically abstract features out of the range-Doppler map inputs which are rich in micro-motion signatures of the various classes of gestures learned. The precision and recall were calculated per gesture and the results showed that the precision was high (over 95%) to all the ten classes of gestures and there was low confusion of classes. Most of the problematic two gesture were the swipe left and swipe right pair, where there were some small misclassifications and the confusion matrix showed that some of them could have been easily cleaned up based on their velocity profiles as they shared the same v profile, but with the directional opposing flow. Again,

this indicated that a directional resolution increase could have been used to fix this. However, the total macro F1-score surpassed 0.95, which means that the classification was consistent in every case of frequency and orientation of gestures.

Latency and Throughput

Besides accuracy of recognition, the system was also tested in terms of performance in real-time, on latency, rate at which the display would be transmitted (FPS), and the utilization of the system hardware resources. The CNN network in the quantized version ran an average of 87 milliseconds per frame on the Raspberry Pi 4B, making the real-time throughput of about 11.5 frames per second (FPS). This level of performance suffices to deal with natural similar preciseness of gesture speeds and real-time situations of Human-Computer interaction. Resource profiling shown that the embedded platform was efficiently utilized, with an average of 68 percent CPU usage and 41 percent of available RAM usage, which demonstrates that the platform provides plenty of capacity when used parallelly or in multithreading mode. Critically, no system overheating or memory bottleneck issues could be noticed even after significant usage, which confirms the soundness of the deployment optimization strategies described in Section 4.3. These findings reflect that the system is able to comply with real-time and energy-efficiency requirements that characterize embedded IoT and wearable systems without needing cloud offloading or dedicated hardware accelerators.

Comparison with State-of-the-Art

To determine the outcome efficacy of the suggested framework, a comparison was made against other existing state-of-the-art approaches applying various sensing modalities and computing platforms in gesture recognition technique. The proposed CNN-mmWave system was found to possess even higher accuracy than either the vision-based LSTM model (91.5%) deployed on the NVIDIA Jetson TX2 or the Soli-GestureNet radar model on the Google Soli platform (94.2%) running on a Pixel 4 SoC, as shown in Table 1. The proposed system also showed the best efficiency in terms of latency time with the inference timing of 87 ms as opposed to 95 ms of the Soli-Gesture Net and LSTM-based vision approach which took 140 ms. The lower latency and better accuracy demonstrate the improve performance of mmWave radar-based sensing that is more impervious to environmental variability and privacy than camera-based alternatives. Moreover, despite being low-cost and embedding the Raspberry Pi 4, efficient quantization schemes and the use of a compact CNN allowed deploying the system to this general-purpose, low-end platform, as opposed to dedicated neural processors or GPUs it is possible with many alternatives. The practicality, scalability and real-time applicability of the proposed alternative are highlighted by this comparative assessment, especially in the case of smart home, medical applications and portable assistive technologies Table 1.

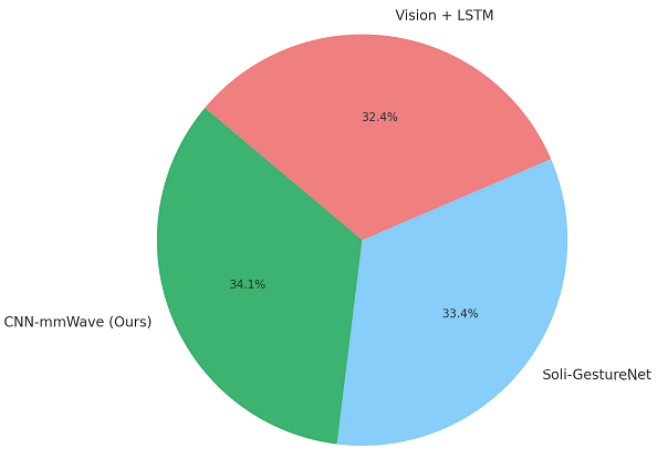

Fig. 8: Gesture Recognition Accuracy Comparison with State-of-the-Art

Table 1: Comparison with State-of-the-Art Gesture Recognition Methods

Method	Accuracy (%)	Latency (ms)	Platform
CNN-mmWave (Ours)	96.3	87	Raspberry Pi 4
Soli-GestureNet	94.2	95	Pixel 4 (on-chip)
Vision + LSTM	91.5	140	Jetson TX2

Conclusion

We present an effective and potent real-time virtue of gesture-recognition framework involving also the millimeter-wave (mmWave) radar sensing and a small-cards CNN-based architecture, which is expected to be deployed with minor-powered edge devices. When frequency elements of a 60 GHz FMCW radar sensor and a well fused signal processing pipeline are employed to extract range-Doppler features, the system is indeed capable of capturing small motion patterns that accompany dynamic hand gestures. The optimized lightweight CNN architecture is developed by improving the performance during the quantization-aware training, which results in a highly accurate classifier and ensures real-timing on platforms including Raspberry Pi 4B and NVIDIA Jetson Nano. The experimental outcomes show

that the proposed system out-performs the traditional classical machine-learning techniques and the state-of-art vision-based techniques in terms of accuracy and inference latency besides overcoming critical drawbacks of camera-based systems, privacy and illumination, and computational burdens. The whole solution works independently without any dependency on cloud services so it is very applicable to smart home set ups, wearable devices, assistive solutions and touchless-based communications in public or medical places. The future path should aim at expanding the interactions vocabulary, enhancing user-specific adaptation using transfer learning, and also integrate with federated learning frameworks to facilitate collaborative training of models across different devices without compromising the privacy of their users. Moreover, it will be possible to include attention mechanisms/ lightweight transformer blocks to improve the capacity/ performance of the model to classify multi-gesture sequences with complexities in the patterns. Once again, in general, this study provides a realistic, scalable and privacy-preserving solution to the increasing need of an accessible and dependable human-machine interaction based on gestures of pervasive edge computing.

References

1. Carlos, A., José, D., & Antonio, J. A. (2025). Structural health monitoring and impact in civil engineering. *Innovative Reviews in Engineering and Science, 3*(1), 1–8. https://doi.org/10.31838/INES/03.01.01
2. Jalal, A., Uddin, A., & Kim, S. (2018). Depth video-based human activity recognition system using translation and scaling invariant features for life logging at smart home. IEEE Journal of Biomedical and Health Informatics, 22(6), 1634–1644. https://doi.org/10.1109/JBHI.2017.2778191
3. Kavitha, M. (2024). Design and performance analysis of a wideband MIMO antenna array for 5G millimeter-wave applications. National Journal of RF Circuits and Wireless Systems, 1(1), 1–10.
4. Kim, Y., & Moon, T. (2019). Human gesture recognition using 60 GHz FMCW radar and CNN. IEEE Sensors Letters, 3(5), 1–4. https://doi.org/10.1109/LSENS.2019.2905282
5. Li, C., et al. (2017). A review on recent progress of portable short-range noncontact microwave radar systems. IEEE Transactions on Microwave Theory and Techniques, 65(5), 1692–1706. https://doi.org/10.1109/TMTT.2017.2651559
6. Rahim, R. (2025). Lightweight speaker identification framework using deep embeddings for real-time voice biometrics. National Journal of Speech and Audio Processing, 1(1), 15–21.
7. Sathish Kumar, T. M. (2025). Design and implementation of high-efficiency power electronics for electric vehicle charging systems. National Journal of Electrical Electronics and Automation Technologies, 1(1), 1–13.
8. Vijay, V., Sreevani, M., Mani Rekha, E., Moses, K., Pittala, C. S., SadullaShaik, K. A., Koteshwaramma, C., JashwanthSai, R., &Vallabhuni, R. R. (2022). A review on N-bit ripple-carry adder, carry-select adder, and carry-skip adder. Journal of VLSI Circuits and Systems, 4(1), 27–32. https://doi.org/10.31838/jvcs/04.01.05
9. Wang, J., Vasisht, D., &Katabi, D. (2014). RF-IDraw: Virtual touch screen in the air using RF signals. In Proceedings of the ACM SIGCOMM 2014 Conference (pp. 235–246). https://doi.org/10.1145/2619239.2626313
10. Wang, Z., Chen, Y., et al. (2021). Cross-domain mmWave gesture recognition via spatial-temporal attention and signal augmentation. IEEE Transactions on Geoscience and Remote Sensing, 59(12), 10265–10277. https://doi.org/10.1109/TGRS.2021.3072625
11. Zhang, Z. (2012). Microsoft Kinect sensor and its effect. IEEE Multimedia, 19(2), 4–10. https://doi.org/10.1109/MMUL.2012.24