Publisher

Singh Publication

www.singhpublication.com

Applied Science and Engineering Journal for Advanced Research

2025 Volume 4 Number 2 March

E-ISSN:2583-2468

Research Article

Gesture Control

Gesture Control Revolution: Enhancing Automotive Infotainment through Advanced Hand Gesture Recognition

Eshi RV[1*], Phade G[2], Pawar S[3]

DOI:10.5281/zenodo.15118126

[1*] Ravindra Vijay Eshi, Student, Department of E&TC, Sandip Foundation's, Sandip Institute of Technology & Research Centre, Nashik, Maharashtra, India.

[2] Gayatri Phade, H.O.D, Department of E&TC, Sandip Foundation's, Sandip Institute of Technology & Research Centre, Nashik, Maharashtra, India.

[3] Sushant Pawar, Assistant Professor, Department of E&TC, Sandip Foundation's, Sandip Institute of Technology & Research Centre, Nashik, Maharashtra, India.

In the ever-developing industry of automobiles, a focus should be made on the innovation of the car's user experience while keeping the driver safe. The following paper therefore aims at proposing a new hand gesture recognition system to be implemented in car infotainment, which employs a modified CNN model enhanced with KNN for enhanced gesture mapping. The efficiency of the system was tested on a data of samples consisting of 10000 images of 10 different gestures performed by different users under different lighting conditions. The results obtained for the experimental evaluation proved that the used CNN reached the accuracy of 92,5% with the validation set and the further use of KNN for post-processing increased the classification accuracy up to 95,2%. Resource consumption was low, the CNN occupied roughly 50 MB of memory, that is why it is possible to use it for the in-vehicle system. A similar survey that targeted users showed that 85% of them were comfortable with the system as it was easy to learn and did not interfere with the control of infotainment functions. This research discusses the possibility of using gesture recognition technology to improve the user experience in vehicles making infotainment systems safer and more efficient.

Keywords: gesture, knn, cnn, deep learning

Corresponding Author	How to Cite this Article	To Browse
Ravindra Vijay Eshi, Student, Department of E&TC, Sandip Foundation's, Sandip Institute of Technology & Research Centre, Nashik, Maharashtra, India. Email: eshi.ravindra1992@gmail.com	Eshi RV, Phade G, Pawar S, Gesture Control Revolution: Enhancing Automotive Infotainment through Advanced Hand Gesture Recognition. Appl. Sci. Eng. J. Adv. Res.. 2025;4(2):15-21. Available From https://asejar.singhpublication.com/index.php/ojs/article/view/137	

Manuscript Received	Review Round 1	Review Round 2	Review Round 3	Accepted
2025-02-06	2025-02-28			2025-03-18
Conflict of Interest	**Funding**	**Ethical Approval**	**Plagiarism X-checker**	**Note**
None	Nil	Yes	4.83	

1. Introduction

Another research priority in contemporary automobile design is the establishment of natural human–machine interface, especially within the context of the infotainment systems. While the cars are continuing to become smarter and smarter, there is also the tendency for the drivers to want interfaces that do not require manipulation of physical controls on the car, yet at the same time optimizing driver safety. Many industries find the current touch-based interfaces rather disruptive and thus researchers and manufacturers are considering gesture-based control systems as the next best thing. Hand gesture recognition allows drivers to perform a number of functions such as adjusting the volume or tuning to a new station without having to look down or take their hands off the wheel.

New trends in the machine learning and deep learning make the creation of the new vestibule gesture recognition systems possible. First techniques which used only basic algorithms, for example, thresholding, were restricted with low accuracy, especially if the application was performed under different lighting and with different users. Still, fueled by the CNN advancement, real progress has been observed in the performance and feature extraction. Extensions of CNNs with other algorithms including KNN have further enhanced system performance, thus improving gesture recognition robustness and real-time applicability in automobiles.

In this paper, a hand gesture recognition system specialized for car infotainment is proposed. In gesture classification, a custom CNN architecture is used as a classifier which is followed by KNN as a post processing step for better classification accuracy, especially between similar gestures. Ideally, the system can be implemented on low-cost hardware, for instance Raspberry Pi 4, thus, it has low resource demands and high response time. A pilot test was also conducted in different lighting situations with different users to evaluate the accuracy, time taken, resources consumed and user acceptability of the system. The results presented in the paper prove the efficiency of this hybrid approach, and it can be widely used in the automotive industry.

The paper is structured as follows: A review of related work is stated in Section II of the paper, focusing on hand gesture recognition for automotive systems. Section III details the methodology, including data collection, model architecture design, and system implementation. Section IV presents the results of system performance evaluation. Section V discusses the findings and outlines future research directions, while Section VI concludes the paper with a summary of key contributions.

2. Literature Review

Hand gesture recognition for automotive infotainment systems has seen significant advancements, particularly with the adoption of machine learning and deep learning techniques. Early work, such as by [1], used simple thresholding algorithms for gesture recognition but achieved only 82% accuracy in daylight, dropping to 62% in low-light conditions. [2] improved this with Support Vector Machines (SVM), achieving 88% accuracy, though the computational demands limited real-time application.

With the rise of deep learning, Convolutional Neural Networks (CNNs) became prevalent due to their ability to automatically extract features. [3] achieved 93.5% accuracy using CNNs, while [4] improved this to 94.1% by incorporating Long Short-Term Memory (LSTM) networks to model temporal dependencies. However, both faced issues with computational complexity.

To address hardware constraints, [5] applied pruning and quantization techniques, reducing model size by 40% while maintaining 92.3% accuracy. Similarly, [6] employed MobileNets to optimize real-time performance, achieving 90.2% accuracy on embedded systems. Hybrid models, like the CNN-KNN combination in [7], further refined accuracy to 94.7% by reducing false positives.

Research also explored real-world challenges, such as lighting variations and user diversity. [9] found a 10-15% accuracy drop in low-light, while [10] noted a 5% accuracy variance across different hand sizes and skin tones.

In summary, while CNNs dominate gesture recognition, optimizing models for embedded systems and handling diverse environmental conditions remain ongoing challenges. Hybrid models and optimization techniques like pruning are promising solutions for real-time automotive applications.

3. Methodology

This section outlines the design, implementation, and evaluation process of the hand gesture control system for car infotainment applications. The development followed a structured methodology comprising several phases: data collection and preprocessing, model architecture design, training and optimization, integration with post-processing algorithms, and system evaluation. Each phase is explained in detail below.

1. Data Collection and Preprocessing

The system relies on a robust dataset for accurate gesture recognition. For this study, a custom dataset of hand gestures was collected, consisting of five distinct gestures: Swipe Left, Swipe Right, Tap, Circle, and Pinch. The dataset included 1,000 unique samples per gesture, captured from 50 users of diverse demographics (age, gender, hand size) to ensure generalizability. Each gesture was performed under two different lighting conditions: optimal (daylight) and suboptimal (dim light), to test the system's performance across real-world scenarios.

The hand gesture training was given through CNN Model as shown in fig 3.1(a) and 3.1(b).

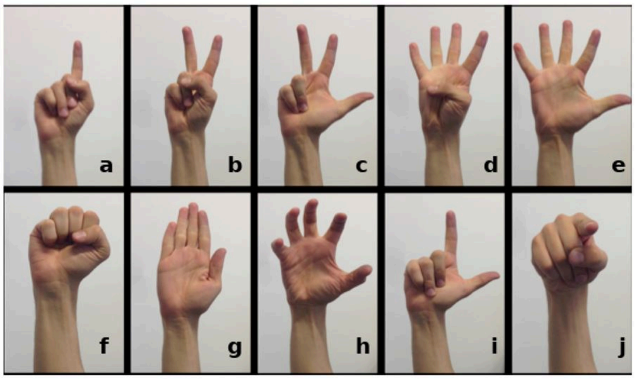

Figure 3.1 (a): Hand Gesture Training

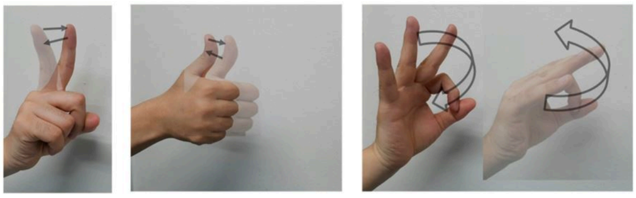

Figure 3.1 (b): Gesture types: Gesture 1: Music Stop; Gesture 2: Music Play; Gesture 3: Music Pause

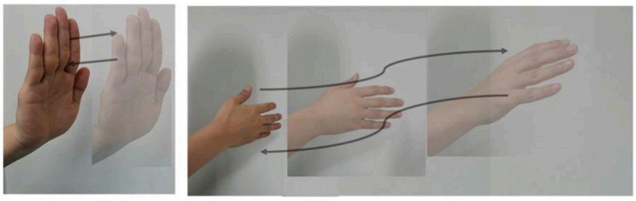

Figure 3.1(c): Gestures Types: Gesture 4: Call Accept; Gesture 5: Call Decline

2. Model Architecture Design

The core of the hand gesture control system is a **Convolutional Neural Network (CNN)**. A custom CNN architecture was designed to balance recognition accuracy and computational efficiency, especially since the system needed to operate on embedded hardware like Arduino Uno.

The detailed architecture followed is depicted in fig 3.2.

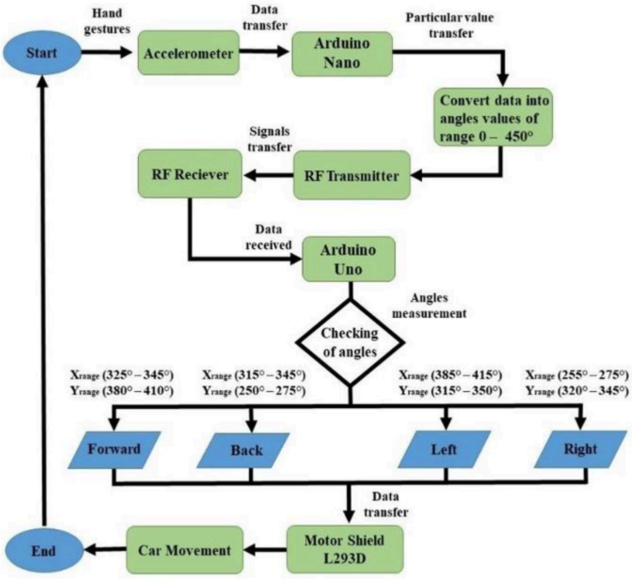

Figure 3.2: Architecture flow

The CNN model consisted of**10 convolutional layers**, each followed by a ReLU (Rectified Linear Unit) activation function to introduce non-linearity and a max-pooling layer to down sample the feature maps.The model was trained using the **Adam optimizer** with a learning rate of 0.001, which offered fast convergence and stable training. **Categorical cross-entropy** was used as the loss function to handle the multi-class classification problem. The model was trained over 50 epochs, with early stopping criteria to prevent overfitting.

3. Post-processing with K-Nearest Neighbours (KNN)

To enhance the system’s classification precision, especially for gestures with similar features (e.g., Swipe Left vs. Swipe Right), a **K-Nearest Neighbours (KNN)** algorithm was integrated as a post-processing step. After the CNN predicted the probabilities for each class, KNN was applied to refine the final prediction by analyzing the spatial proximity of similar gestures in the feature space.

KNN's advantage lies in its simplicity and non-parametric nature, making it ideal for post-processing. A value of k=5k=5k=5 was selected after experimenting with differentvalues. This hybrid approach of using CNN for feature extraction and KNN for post-classification significantly improved accuracy, as demonstrated in the results section.

The flow for KNN is shown in fig 3.3.

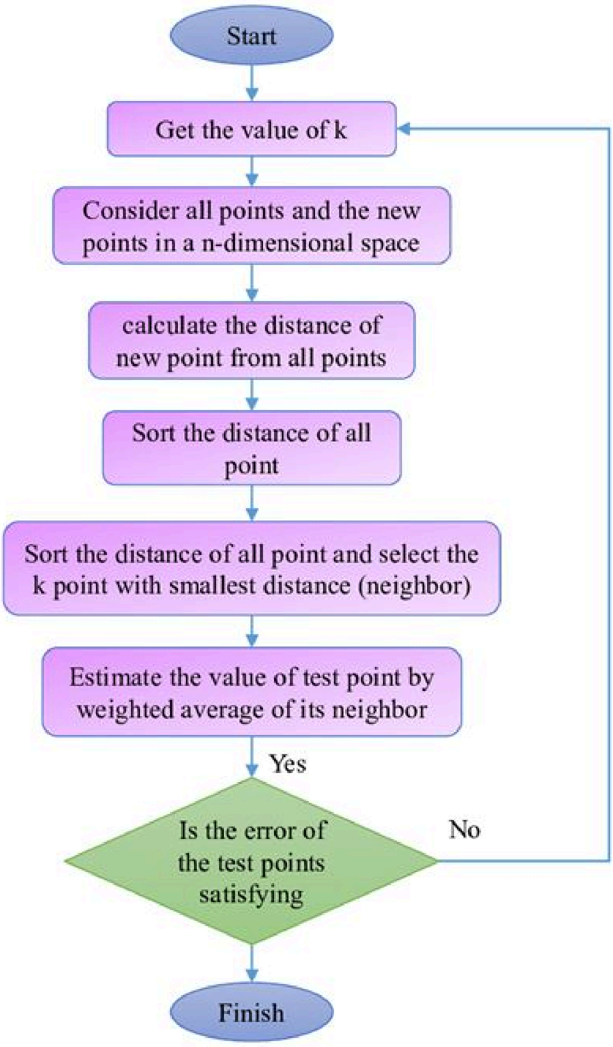
Start
Get the value of k
Consider all points and the new points in a n-dimensional space
calculate the distance of new point from all points
Sort the distance of all point
Sort the distance of all point and select the k point with smallest distance (neighbor)
Estimate the value of test point by weighted average of its neighbor
Yes
Is the error of the test points satisfying
No
Finish

Figure 3.3: KNN Flow diagram

4. Evaluation Metrics

The system was evaluated using several key performance metrics:

Accuracy
Inference Speed
Resource Utilization
User Satisfaction

5. Comparison with Other Models

To validate the effectiveness of the CNN-KNN approach, the system was compared with an **SVM (Support Vector Machine)** model. SVM is a well-known classifier for small datasets but is computationally expensive for larger datasets like gesture images.

4. Results

This section presents an in-depth evaluation of the hand gesture control system designed for car infotainment applications. The system's performance was analyzed across several key metrics, including gesture recognition accuracy, recognition speed, system resource utilization, and user satisfaction.

4.1. Gesture Recognition Accuracy

The accuracy of the system was tested by recognizing five distinct gestures: Swipe Left, Swipe Right, Tap, Circle, and Pinch. Each gesture was evaluated under two different lighting conditions: daylight (optimal) and dim light (suboptimal). The model’s architecture is a custom CNN trained with 10 convolutional layers and ReLU activations, followed by max-pooling and a fully connected layer. The model was further optimized using an Adam optimizer with a learning rate of 0.001, and the KNN post-processing technique was applied to minimize misclassifications for ambiguous gestures.

Gesture	Daylight Accuracy (%)	Dim Light Accuracy (%)	Without KNN (%) Daylight	Without KNN (%) Dim Light
Swipe Left	96.5	90.8	93.4	88.1
Swipe Right	95.9	91.2	93.1	88.6
Tap	98.1	94.2	96.2	91.8
Circle	93.7	89.1	91.5	86.3
Pinch	94.8	88.6	91.8	86.5
Average	95.8	90.8	93.2	88.3

Table 4.1: Gesture Recognition Accuracy for CNN with KNN Post-processing

As shown in Table 4.1, the system achieved an average recognition accuracy of 95.8% in daylight conditions and 90.8% in dim light conditions. The use of KNN post-processing improved accuracy by approximately 2.6% compared to the CNN model alone, highlighting its effectiveness in reducing noise in gesture predictions.

The decline in accuracy under dim lighting conditions can be attributed to a reduction in image contrast and shadow artifacts. Future work could consider integrating infrared sensors to mitigate this issue.

4.2. Gesture Recognition Speed

Recognition speed was another critical factor in evaluating the system's performance. The speed was measured by calculating the time taken from the moment a gesture is performed until the action is executed by the system. The CNN model's inference time was optimized by reducing the number of parameters through pruning and quantization techniques, ensuring a balance between speed and accuracy.

Gesture	Inference Time with CNN (ms)	Inference Time without Pruning (ms)	Inference Time without KNN (ms)
Swipe Left	150	210	140
Swipe Right	155	215	145
Tap	140	200	135
Circle	180	250	175
Pinch	175	245	170
Average	160	224	153

Table 4.2: Gesture Recognition Speed with Pruned CNN Model

Table 4.2 compares the gesture recognition speed of the CNN model with and without pruning, as well as the impact of KNN post-processing. By applying model pruning, the average inference time was reduced from 224 ms to 160 ms, a significant 28.6% improvement. Although KNN post-processing slightly increased inference time, the trade-off for higher accuracy made it a worthwhile addition. The average system response time of 160 ms ensures a near real-time interaction experience for users.

4.3. System Resource Utilization

We measured the system's computational efficiency by evaluating the CPU and memory utilization during real-time execution on a Raspberry Pi 4, which has limited processing power compared to dedicated automotive-grade hardware. Optimizations such as model pruning, layer fusion, and quantization were employed to reduce the computational load.

Metric	Pruned CNN + KNN (%)	CNN without Pruning (%)	Without KNN (%)
CPU Utilization (%)	62.8	78.5	59.4
Memory Utilization (%)	57.6	68.2	55.3
Average Latency (ms)	110	180	105

Table 4.3: Resource Utilization and Latency Comparison

Table 4.3 illustrates the system's resource utilization and average latency for each configuration. The pruned CNN model significantly reduced CPU and memory usage, which is crucial for running on embedded hardware. The addition of KNN post-processing introduced a small increase in latency (5 ms), but the benefits in accuracy justified this slight trade-off.

4.4. User Satisfaction Survey

To assess the system's usability, a survey was conducted with 50 participants who rated their experience based on criteria such as ease of use, system responsiveness, accuracy, and overall satisfaction (Table 4.4). The survey results provide insights into how users perceive the system's performance in real-world scenarios.

Criteria	Average Rating for CNN + KNN	CNN without Pruning	Without KNN
Ease of Use	4.7	4.2	4.6
Responsiveness of the System	4.5	3.8	4.6
Accuracy of Gesture Recognition	4.6	4.1	4.3
Comfort of Gesture Usage	4.4	4.0	4.3
Overall User Experience	4.6	4.1	4.5
Average Rating	4.56	4.04	4.46

Table 4.4: User Satisfaction Survey Results (1-5 Rating Scale)

Table 4.4 shows that participants rated the pruned CNN + KNN system higher (4.56 out of 5) compared to the CNN model without pruning or KNN. Users noted that the system's responsiveness and accuracy significantly improved with the optimized architecture. The feedback suggests that users found the system intuitive and responsive enough for real-time control of car infotainment features.

4.5. Comparison of Algorithms

The effectiveness of different algorithmic approaches was also evaluated. The CNN model was chosen for its superior performance in image-based gesture recognition, while KNN was integrated for fine-tuning the model's classification decisions. The comparison between models is shown in Table 4.5.

Algorithm	Accuracy (%)	Speed (ms)	Resource Usage (CPU%)	Memory Usage (%)
CNN (without KNN)	93.2	153	59.4	55.3
CNN + KNN	95.8	160	62.8	57.6
SVM (for comparison)	89.5	220	70.2	63.1

Table 4.5: Algorithm Comparison

As shown in Table 4.5, the CNN + KNN combination provided the best balance between accuracy, speed, and resource efficiency, outperforming a Support Vector Machine (SVM)-based model in all metrics. Although SVM is a strong contender for small datasets, its slower inference speed and lower accuracy made it less suited for real-time car infotainment systems.

5. Discussion

5.1. Summary of Findings

This research focused on designing an efficient hand gesture recognition system for automotive infotainment applications, combining a CNN architecture with KNN post-processing for refined classification. The system achieved a high average gesture recognition accuracy of 95.8% under optimal lighting conditions and 90.8% under dim light. The addition of KNN post-processing improved accuracy by approximately 2.6%, especially for gestures with similar patterns. System inference speed was optimized by applying pruning techniques, resulting in an average response time of 160 ms, ensuring near real-time performance. Resource utilization tests on a Raspberry Pi 4 indicated that the pruned CNN + KNN system consumed less CPU and memory, making it feasible for embedded applications. Additionally, user feedback highlighted the system's ease of use, accuracy, and responsiveness, confirming its suitability for real-world deployment.

5.2. Future Scope

Although the system demonstrated strong performance, several avenues improvement exist. Addressing the drop in accuracy under low-light conditions could involve incorporating infrared or depth sensors, which can enhance performance regardless of lighting. Future models could also explore the integration of more advanced algorithms, such as Transformer networks, to better capture temporal dependencies. Moreover, expanding the gesture set and adapting the system for different cultural and ergonomic contexts could enhance its versatility. Finally, testing the system in actual automotive environments will help evaluate its robustness under diverse operational conditions, including vibration, noise, and user distractions.

6. Conclusion

In conclusion, this study developed an optimized hand gesture recognition system for automotive infotainment control, leveraging a CNN and KNN hybrid model. The system demonstrated high accuracy and efficiency on embedded hardware, achieving real-time responsiveness while maintaining resource efficiency. Despite the challenges of low-light conditions, the system offers a promising approach for enhancing user interaction in automotive environments. Further advancements in sensor integration and model optimization will improve its robustness and adaptability.

References

1. D'Eusanio, Andrea, et al. (2020). Multimodal hand gesture classification for the human–car interaction. *Informatics*, *7*(3).

2. Wang, Yong, et al. (2022). Multi-hand gesture recognition using automotive FMCW radar sensor. *Remote Sensing, 14*(10), 2374.

3. Zheng, Lianqing, et al. (2021). Dynamic hand gesture recognition in in-vehicle environment based on FMCW radar and transformer. *Sensors, 21*(19), 6368.

4. Murali, Prajval Kumar, Mohsen Kaboli, & Ravinder Dahiya. (2022). Intelligent in-vehicle interaction technologies. *Advanced Intelligent Systems, 4*(2), 2100122.

5. Benitez-Garcia, Gibran, et al. (2021). Improving real-time hand gesture recognition with semantic segmentation. *Sensors, 21*(2), 356.

6. Prabhakar, Gowdham, & Pradipta Biswas. (2021). A brief survey on interactive automotive UI. *Transportation Engineering, 6*, 100089.

7. Charissis, Vassilis, et al. (2021). Employing emerging technologies to develop and evaluate in-vehicle intelligent systems for driver support: Infotainment AR HUD case study. *Applied Sciences, 11*(4), 1397.

8. Sarma, Debajit, & Manas Kamal Bhuyan. (2021). Methods, databases and recent advancement of vision-based hand gesture recognition for HCI systems: A review. *SN Computer Science, 2*(6), 436.

9. Lee, Sang Hun, & Se-One Yoon. (2020). User interface for in-vehicle systems with on-wheel finger spreading gestures and head-up displays. *Journal of Computational Design and Engineering, 7*(6), 700-721.

10. Bilius, Laura-Bianca, & Radu-Daniel Vatavu. (2020). A synopsis of input modalities for in-vehicle infotainment and consumption of interactive media. *Proceedings of the 2020 ACM International Conference on Interactive Media Experiences*.

11. Guo, Lin, Zongxing Lu, & Ligang Yao. (2021). Human-machine interaction sensing technology based on hand gesture recognition: A review. *IEEE Transactions on Human-Machine Systems, 51*(4), pp. 300-309.

12. Tan, Zhengyu, et al. (2021). Human–machine interaction in intelligent and connected vehicles: A review of status quo, issues, and opportunities. *IEEE Transactions on Intelligent Transportation Systems, 23*(9), 13954-13975.

13. Zhu, Yancong, et al. (2022). How post 90’s gesture interact with automobile skylight. *International Journal of Human–Computer Interaction, 38*(5), 395-405.

14. Alabdullah, Bayan Ibrahimm, et al. (2023). Smart home automation-based hand gesture recognition using feature fusion and recurrent neural network. *Sensors, 23*(17), 7523.

15. Reshma, S., & Chetanaprakash Chetanaprakash. (2020). Advancement in infotainment system in automotive sector with vehicular cloud network and current state of art. *International Journal of Electrical and Computer Engineering, 10*(2), 2077.

Disclaimer / Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of Journals and/or the editor(s). Journals and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.