Journal of Machine Learning and Deep Learning (JMLDL)
Volume 3, Issue 1, Date: 30- June – 2026
ISSN: 2978-4336, DOI: 10.64820/AEPJMLDL.31.69.72.62026

Real-Time Gesture Recognition Using Convolutional Neural Networks (CNN)

Heena Khera [1], Ahsash Singh [2, *], Abhishek Kumar Yadav [3], Abhinandan Malviya [4], Aviral Srivastava [5], Anamika Pandey [6]
[1, 2, 3, 4,5, 6] School of Computer Science & Engineering, IILM University, Greater Noida, India
Email: [1] ahsashsingh24@gmail.com
*Corresponding Author

***Abstract*—This paper presents a vision-based gesture recognition system trained using a Convolutional Neural Network (CNN). Gesture data is collected via a standard RGB webcam, pre-processed through grayscale conversion, background subtraction, Gaussian blur, and pixel normalisation, and then fed into a three-block CNN for feature extraction and classification. The proposed model is trained with the Adam optimiser (lr = 0.001, batch size = 32) using categorical cross-entropy loss over 50 epochs with early stopping. On a 10-class custom dataset of 5,000 labelled images, the system achieves an overall macro-average accuracy of 95.5%, with real-time inference running at approximately 28 fps on commodity hardware. Comparative evaluation against state-of-the-art vision-based methods confirms competitive performance without requiring specialised sensors.**

Keywords—Convolutional Neural Network (CNN); Gesture Recognition; Human-Computer Interaction; Image Processing; Deep Learning; Real-Time Classification; Sign Language.

I. INTRODUCTION

Human-Computer Interaction (HCI) has evolved dramatically over the past decade. Traditional input devices such as keyboards and mice impose rigid constraints on how users communicate with machines, limiting accessibility, expressiveness, and intuitiveness. Gesture recognition — the ability of a computer system to interpret meaningful human body movements — has emerged as a compelling alternative that promises a more natural and efficient interface [1].

Gesture recognition systems translate movements of the hands, face, or body into computer commands. Applications span a wide spectrum: from enabling deaf individuals to communicate through sign language recognition systems, to enhancing immersive experiences in gaming and virtual reality, to supporting surgeons in sterile operating environments where touchless control of displays is essential [2].

Early gesture recognition approaches relied on wearable hardware sensors such as data gloves, which captured joint angles and finger positions. While accurate, such systems were expensive, cumbersome, and impractical for everyday use. The proliferation of affordable, high-resolution cameras and the advent of deep learning have together catalysed a shift towards vision-based approaches, which require no special hardware and operate directly on image or video input [3].

Convolutional Neural Networks (CNNs) have established themselves as the dominant architecture for image-based gesture recognition. Their ability to automatically learn hierarchical spatial features from raw pixel data — without manual feature engineering — has yielded state-of-the-art results on multiple benchmark datasets. Recent work has further integrated Long Short-Term Memory (LSTM) networks to capture temporal dynamics in video sequences [4].

This paper presents a CNN-based real-time gesture recognition system developed at IILM University, Greater Noida. The system collects gesture data via a standard RGB camera, preprocesses frames, extracts features using a three-block CNN, and classifies gestures in real time. The remainder of this paper is organised as follows: Section II reviews related literature technology-by-technology; Section III describes the proposed methodology with mathematical formulations; Section IV presents experimental results and analysis; Section V concludes the work.

II. LITERATURE REVIEW

This section surveys key contributions to vision-based, ML-driven gesture recognition, presented technology by technology.

A. Convolutional Neural Network (CNN) for Static Gesture Recognition

CNNs have become the cornerstone of static gesture recognition by automatically learning spatial features from image data. Barbhuiya et al. demonstrated that CNN-based feature extraction significantly outperforms hand-crafted descriptors for sign language classification, reporting improved accuracy on benchmark datasets with architectures such as VGG and DenseNet [5]. Niu further designed a CNN-based HCI system specifically for real-time gesture-driven control, demonstrating robust performance across varying lighting conditions and confirming the feasibility of deploying CNN models in resource-constrained environments without sacrificing accuracy [7].

B. 3D-CNN and LSTM for Dynamic Gesture Recognition

Dynamic gestures require modelling temporal sequences of hand movements. Chen et al. proposed a hybrid 3D-CNN + LSTM framework for static/dynamic gesture-driven human-robot control, achieving 93.18% validation accuracy

Received: 17- 4 - 2026
Revised: 30-6-2026
Published: 30-6-2026

on a custom dataset [10]. The 3D-CNN captures short-range spatiotemporal features within small frame windows, while the LSTM module models long-range temporal dependencies. Lee et al. further demonstrated CNN-LSTM pipelines in 3D virtual reality gaming applications, confirming practical viability in interactive entertainment contexts [9].

C. Star-RGB Temporal Condensation with CNN

Santos et al. proposed recognising dynamic gestures using CNNs combined with a Star-RGB representation — a temporal condensation technique that encodes motion across multiple frames into a single composite image [8]. This approach achieved competitive accuracy of 92.5% without the computational overhead of recurrent architectures, making it suitable for deployment on hardware with limited memory and processing capability.

D. Attention Mechanisms and Transformer-Based Architectures

Attention mechanisms have been incorporated to direct model focus onto gesture-relevant spatial and temporal regions. The CNN-AttST model integrates spatiotemporal attention with CNN for recognising complex muscle activation patterns in EMG-based gesture data, outperforming conventional models on the Ninapro DB2 dataset [11]. Vision Transformers (ViTs) model long-range spatial dependencies via self-attention and have shown competitive performance in gesture classification benchmarks, although at higher computational cost relative to lightweight CNNs [4].

E. Lightweight CNN on Thermal Video Data

To overcome the sensitivity of RGB cameras to lighting variation and skin-tone differences, Birkeland et al. proposed a lightweight CNN operating on thermal video sequences, achieving 97% classification accuracy on a dataset of 15-frame-per-gesture sequences [12]. Thermal imaging provides illumination-invariant input, substantially improving recognition robustness in uncontrolled environments. This advantage comes at the cost of requiring specialised thermal cameras, which motivates the multi-modal fusion research direction.

F. Multi-Modal Fusion: RGB + Depth (RGBD)

Multi-modal fusion of RGB and depth information has improved gesture recognition robustness in occluded and cluttered backgrounds [3]. RGBD approaches exploit depth maps from sensors such as Intel RealSense to segment the hand from complex backgrounds more reliably than RGB-only methods. Leonardi et al. comprehensively reviewed deep learning approaches for hand detection, segmentation, and gesture recognition in human-robot interaction, highlighting RGBD as the leading fusion strategy for industrial deployments [3].

G. Transfer Learning and Large Pretrained Models

Transfer learning from large pretrained models such as MediaPipe Hands, MobileNet, and ResNet has substantially reduced the data requirements for gesture recognition. Arooj et al. applied CNN combined with SIFT for Pakistan Sign Language recognition, demonstrating that feature-level fusion of handcrafted and learned descriptors improves accuracy when training data is limited [13]. Zhen et al. extended this paradigm to cross-session EMG-based gesture recognition using deep learning and transfer learning, achieving robust performance despite inter-session variability [14].

H. Wearable and Sensor-Based Gesture Recognition

Beyond vision, machine-learned wearable sensors have enabled real-time hand-motion recognition for practical applications. Yeo et al. demonstrated flexible, skin-integrated sensors combined with ML classifiers for real-time gesture decoding, opening pathways for prosthetics, rehabilitation, and AR/VR control without any camera infrastructure [15]. This strand of research complements vision-based approaches by serving scenarios where cameras are impractical or privacy-sensitive.

III. Proposed Methodology

A. System Architecture

The proposed system follows a four-stage pipeline: (1) Data Collection, (2) Pre-processing, (3) Feature Extraction via CNN, and (4) Classification and Output. A standard RGB webcam serves as the input device, making the system accessible without specialised hardware.

B. Data Collection

Gesture data is captured at 30 fps from a standard RGB webcam. Each of the 10 gesture classes comprises a minimum of 500 labelled samples recorded under varying backgrounds, lighting conditions, and hand orientations. Both static hand poses (e.g., Open Palm, Thumbs Up) and simple dynamic movements (e.g., Swipe Left, Circle) are included.

C. Pre-processing

Raw frames undergo the following sequential pre-processing steps before being fed into the CNN:

- Resize to 128×128 pixels for uniform input dimensions.
- Convert to grayscale or HSV colour space to isolate skin regions.
- Background subtraction using Gaussian Mixture Models (GMM): a pixel x is modelled as a weighted sum of K Gaussian distributions: $P(x) = \Sigma^{k}\ \omega k \cdot N(x; \mu k, \Sigma k)$, k = 1, 2, …, K — where ωk, μk, and Σk are the weight, mean, and covariance of the k-th Gaussian component. Pixels whose probability P(x) falls below a threshold θ are classified as foreground.
- Gaussian blur for noise reduction: $I_{blur} = I * G\sigma$, where $G\sigma(x, y) = (1 / 2\pi\sigma^2) \cdot \exp(-(x^2 + y^2) / 2\sigma^2)$.
- Pixel-value normalisation to [0, 1]: $I_{norm}(x, y) = I(x, y) / 255$.

D. CNN Architecture and Feature Extraction

The core of the system is a three-block CNN that jointly performs feature extraction and classification. Let the input frame be $I \in \mathbb{R}^{128\times128\times1}$. Each convolutional layer l applies F_l filters of size 3×3 with ReLU activation: $Z^{l}_{f} = \mathrm{ReLU}(W^{l}_{f} * A^{l-1} + b^{l}_{f})$. Following each

convolutional layer, max-pooling with a 2×2 window and stride 2 halves the spatial dimensions.

Table 1. Proposed CNN Architecture

Layer	Configuration	Output Size	Activation
Input	128×128×1 (grayscale)	128×128×1	—
Conv1	32 filters, 3×3, stride 1	128×128×32	ReLU
MaxPool1	2×2, stride 2	64×64×32	—
Conv2	64 filters, 3×3, stride 1	64×64×64	ReLU
MaxPool2	2×2, stride 2	32×32×64	—
Conv3	128 filters, 3×3, stride 1	32×32×128	ReLU
MaxPool3	2×2, stride 2	16×16×128	—
Flatten	—	32,768	—
FC1 + Dropout	256 units, p = 0.5	256	ReLU
FC2 (Output)	N classes	N	Softmax

E. Loss Function and Optimisation

The network is trained by minimising categorical cross-entropy loss L over the training set of N samples: $L = -(1/N) \cdot \Sigma i \, \Sigma j \, yij \cdot \log(\hat{y}ij)$, where $yij \in \{0,1\}$ is the one-hot true label and $\hat{y}ij = \text{Softmax}(zj)$ is the predicted probability. The model is optimised using the Adam optimiser with initial learning rate $\alpha = 0.001$, $\beta 1 = 0.9$, $\beta 2 = 0.999$, and $\varepsilon = 10^{-8}$. Training proceeds for up to 50 epochs with batch size 32 and early stopping (patience = 10 epochs).

F. Data Augmentation

To improve generalisation, the training set is augmented on-the-fly with:

- Random horizontal flipping (p = 0.5).
- Random rotation in [−15°, +15°].
- Brightness jitter in [−20%, +20%].

G. Real-Time Inference

For real-time inference, OpenCV captures webcam frames at 30 fps. Each frame is passed through the pre-processing pipeline before being classified by the trained CNN. A 5-frame sliding window smooths the class probability distribution: $\hat{y}\_smoothed = (1/T) \cdot \Sigma\_\{t=1\}\^\{T\} \, \hat{y}\_t$, $T = 5$. The predicted gesture label is displayed as an overlay on the live video feed with a confidence score. The pipeline achieves approximately 28 fps on a mid-range CPU.

IV. Experimental Results and Discussion

A. Dataset

Experiments are conducted on two configurations: (i) a custom dataset of 10 gesture classes collected in the laboratory, comprising 5,000 images (500 per class); and (ii) the publicly available NUS Hand Gesture Dataset II (NUS-II) for comparative benchmarking. The dataset is partitioned 80:10:10 (train:validation:test). All images are resized to 128×128 pixels prior to training.

B. Performance Metrics

The system is evaluated using Precision, Recall, and F1-Score: Precision = TP / (TP + FP), Recall = TP / (TP + FN), F1 = 2 · (P · R) / (P + R). Table II summarises per-class performance.

Table 2. Classification Performance on Custom Dataset (10 Classes)

Gesture Class	Precision (%)	Recall (%)	F1-Score (%)	Support
Open Palm	97.2	96.8	97.0	500
Closed Fist	96.5	97.1	96.8	500
Thumbs Up	98.0	97.5	97.7	500
Peace Sign	95.8	96.2	96.0	500
Pointing	96.1	95.7	95.9	500
OK Sign	97.4	97.0	97.2	500
Wave	94.3	93.9	94.1	500
Swipe Left	93.7	94.2	93.9	500
Swipe Right	94.0	93.5	93.7	500
Circle	92.8	92.4	92.6	500
Overall (Macro Avg)	**95.6**	**95.4**	**95.5**	**5000**

C. Comparison with State-of-the-Art

Reference	Method	Dataset	Accuracy (%)
[7]	CNN-HCI System	Custom RGB	94.20
[10]	3D-CNN + LSTM	Custom RGB	93.18
[12]	Lightweight CNN (Thermal)	Custom Thermal	97.00
[8]	CNN + Star-RGB	Custom Video	92.50
Proposed	**Three-Block CNN (RGB)**	**Custom RGB (10-class)**	**95.50**

D. Discussion

Static gestures (Open Palm, Closed Fist, Thumbs Up, OK Sign) consistently achieve precision and recall above 96%,

demonstrating the CNN's strong ability to learn discriminative spatial features from well-defined hand poses. Dynamic gestures (Swipe Left, Swipe Right, Circle, Wave) exhibit lower performance (F1: 92.6–94.1%), which is expected because static-frame analysis cannot fully capture the temporal trajectory information intrinsic to motion-based classes.

The slight accuracy gap relative to thermal-based methods [12] (97.00% vs 95.50%) is attributable to the inherent sensitivity of RGB cameras to illumination variation. Multi-modal RGB+Depth fusion, or the addition of optical flow features as temporal cues, represents a natural extension to close this gap.

Real-time inference runs at approximately 28 fps on a mid-range CPU, confirming practical usability without requiring GPU acceleration or dedicated hardware.

V. Conclusion And Future Work

This paper presented a real-time vision-based gesture recognition system using a three-block Convolutional Neural Network. The system captures gesture data via a standard RGB camera, applies a five-step pre-processing pipeline grounded in Gaussian Mixture Model-based background subtraction and Gaussian blur, and trains a CNN optimised with the Adam algorithm and categorical cross-entropy loss. Experimental evaluation on a 10-class custom dataset of 5,000 images yielded an overall macro-average accuracy of 95.5%, with competitive performance relative to recent state-of-the-art methods and real-time throughput of approximately 28 fps.

Future work will pursue three directions:

- Integration of LSTM layers or optical flow to better model temporal dependencies in dynamic gestures.
- Deployment on embedded platforms (Raspberry Pi, Jetson Nano) to enable edge-based real-time recognition.
- Expansion of the gesture vocabulary to include full sign language alphabets and sentences, leveraging transfer learning from large pretrained models such as MediaPipe Hands.

Acknowledgment

The authors gratefully acknowledge the administrative support of IILM University, Greater Noida, for providing the computational resources and laboratory facilities necessary for this research.

References

[1] T. Starner and A. Pentland, "Real-time American Sign Language recognition from video using hidden Markov models," in Proc. Int. Symp. Computer Vision, 1995, pp. 265–270.

[2] M. Wu, Y. Zhang, J. Li, and H. Wang, "Gesture Recognition Based on Deep Learning: A Review," EAI Endorsed Trans. e-Learning, vol. 11, Mar. 2024.

[3] G. Leonardi et al., "A review on deep learning for vision-based hand detection, hand segmentation and hand gesture recognition in human–robot interaction," Robot. Comput.-Integr. Manuf., vol. 94, Sep. 2025.

[4] R. Aslam, L. Fjeldvik, N. Noori et al., "Deep vision-based real-time hand gesture recognition: a review," PeerJ Comput. Sci., 2025.

[5] A. A. Barbhuiya, R. K. Karsh, and R. Jain, "CNN based feature extraction and classification for sign language," Multimed. Tools Appl., vol. 80, pp. 1–19, 2020.

[6] L. Rossi et al., "Visual Hand Gesture Recognition with Deep Learning: A Comprehensive Review," arXiv:2507.04465, Jul. 2025.

[7] P. Niu, "Convolutional neural network for gesture recognition human–computer interaction system design," PLOS ONE, vol. 20, no. 2, Feb. 2025.

[8] C. C. d. Santos, J. L. A. Samatelo, and R. F. Vassallo, "Dynamic gesture recognition by using CNNs and star RGB," Neurocomputing, vol. 400, pp. 238–254, 2020.

[9] B. H. Lee, T. Kim, and D. H. Oh, "3D Virtual Reality Game with Deep Learning-based Hand Gesture Recognition," J. Korea Comput. Graph. Soc., vol. 24, pp. 41–48, 2018.

[10] C.-S. Zhang et al., "Research on Deep Learning-Based Human–Robot Static/Dynamic Gesture-Driven Control Framework," Appl. Sci., vol. 15, Dec. 2025.

[11] X. Wang and Y. Liu, "Hand Gesture recognition using convolutional neural network and attention mechanism," in Proc. ACM SSIP, 2024.

[12] S. Birkeland et al., "Thermal video-based hand gestures recognition using lightweight CNN," J. Ambient Intell. Humaniz. Comput., vol. 15, pp. 3849–3860, 2024.

[13] S. Arooj et al., "Enhancing sign language recognition using CNN and SIFT," J. King Saud Univ. — Comput. Inf. Sci., vol. 36, p. 101934, 2024.

[14] Z. Zhen et al., "Online cross session electromyographic hand gesture recognition using deep learning and transfer learning," Eng. Appl. Artif. Intell., vol. 127, 2024.

[15] W. H. Yeo et al., "Machine-learned wearable sensors for real-time hand-motion recognition," Natl. Sci. Rev., vol. 11, p. nwad298, 2024.