WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

HAND GESTURE DETECTION AND RECOGNITION USING A CONVOLUTIONAL NEURAL NETWORK

THIRISHA.V

Department of Computer Science and Engineering, SRM, Chennai, India
Corresponding author: thrishamohan2020@gmail.com

ABSTRACT :

RGB hand gesture recognition can support touchless controls, but a classifier evaluated only on manually cropped images does not show how often a webcam system will miss a gesture. This report proposes a two-stage pipeline that uses MediaPipe Hand Landmarker to locate hands and a compact convolutional neural network (CNN) to classify RGB hand crops. The proposed study uses the subject-disjoint splits of the original HaGRID benchmark to evaluate 18 static gestures and an additional class for visible hands without a target gesture. Two tests separate classification on annotated crops from end-to-end recognition on full images. The first uses macro F1 to summarize performance across classes; the second requires both a correct gesture label and adequate overlap with the reference hand box. A separate webcam protocol measures processing latency, throughput, and false activations in idle scenes. Published benchmark results establish context for the design. The proposed pipeline has not yet been evaluated, so its accuracy, operating speed, and suitability for live use remain open questions.

Keywords—hand gesture detection; static gesture recognition; convolutional neural network; MediaPipe; HaGRID; human–computer interaction.

1. Introduction

Static hand poses can provide touchless input for actions such as pausing playback. A single-camera interface must find each hand, recognize its pose, and avoid triggering commands on idle hands. The proposed study asks how much recognition performance changes when a compact CNN receives crops from an automatic hand localizer instead of annotated hand boxes. It evaluates predefined static poses and includes a plan for live webcam testing.

2. Related Work

Freeman and Roth [1] classified gestures using histograms of local image orientation. Pinto et al. [2] combined hand segmentation with a CNN for static American Sign Language letters. Zhang et al. [3] described a palm detector followed by a 21-landmark estimator, which can locate the input region for a separate classifier.

Kapitanov et al. [4] evaluated both classifiers and detectors on HaGRID, a dataset for device-control gestures. MobileNetV3-Small [5] is the lightweight CNN baseline for this study. Molchanov et al. [6] used a recurrent 3D CNN for dynamic gestures in multimodal video, a task that requires temporal evidence and different inputs.

3. Materials and Methods

3.1 System Overview

OpenCV will read test images or capture webcam frames. MediaPipe Hand Landmarker will locate hands, and the CNN will classify each hand crop. A gesture is emitted only when the highest-scoring class is a target gesture and its score meets a validation-selected threshold. The system records the box, class, and score for each emitted gesture.

3.2 Data and Gesture Classes

The planned benchmark is the 554,800-image, 18-gesture HaGRID release described at WACV 2024 [4, 7]. It comprises RGB images from 37,583 contributors. The published split contains 410,800 training images, 54,000 validation images, and 90,000 test images, with no contributor shared across splits. The release annotations contain 3,000 validation images and 5,000 test images per target gesture. These counts describe source images; usable hand crops are counted separately.

The 18 target labels are call, dislike, fist, four, like, mute, ok, one, palm, peace, peace_inverted, rock, stop, stop_inverted, three, three2, two_up, and two_up_inverted. An annotated passive hand contributes a separate no_gesture crop. These boxes can occur in an image that also contains a target gesture, so they cannot serve as empty-frame negatives. The split manifest should identify image IDs, anonymized contributor IDs, and hand counts per class after validation of the annotations.

3.3 Preprocessing and Hand Detection

Convert OpenCV images from BGR to RGB. Use MediaPipe IMAGE mode for independent test images and VIDEO mode with increasing timestamps for webcam frames. Allow up to two hands and set detection, presence, and tracking thresholds to 0.50 [8]. The 21 landmarks define a tight box, clipped to the image for localization scoring. For CNN input, add a 10% margin per side to each detected or annotated box, clip, resize with aspect ratio preserved, and zero-pad to 128 × 128 pixels. Scale RGB values to [0, 1].

Apply rotations of up to 10 degrees and brightness multipliers from 0.8 to 1.2 to training crops only. Avoid mirroring or large rotations because orientation helps distinguish some classes. The CNN receives RGB pixels; landmark coordinates only determine the crop.

3.4 Gesture Recognition

The proposed CNN has three blocks with 32, 64, and 128 filters. Each block uses a 3 × 3 convolution with stride 1 and same padding, followed by batch normalization, ReLU, and 2 × 2 max pooling with stride 2. Global average pooling is followed by a 128-unit ReLU layer, dropout at 0.30, and a 19-class softmax. The model will be trained from scratch on annotated hand crops using categorical cross-entropy and Adam (initial learning rate 0.001; batch size 64; maximum 40 epochs). Select the checkpoint with the highest validation macro F1, stopping after six epochs without improvement.

MobileNetV3-Small [5] will be trained from scratch with the same crop size, data, augmentation, and checkpoint criterion. The architecture's required input normalization will be recorded. For each model, choose the operating threshold on full validation images by maximizing end-to-end macro F1 over the 18 target gestures on a grid from 0.00 to 0.95 in 0.05 increments; break ties in favor of the higher threshold. Freeze the threshold before test evaluation.

3.5 Implementation Environment

The proposed software stack is Python, OpenCV for capture and image preparation, MediaPipe Tasks for hand localization, and TensorFlow/Keras for training and inference. Record the operating system, processor, accelerator, memory, library versions, inference backend, thread count, and MediaPipe model bundle version and checksum for the actual runs. For the webcam test, request 640 × 480 at 30 frames per second and log the resolution and rate the device actually delivers.

4. Experimental Protocol and Evaluation

Verify that contributor IDs are disjoint and keep all crops from the same image in the same split. Use validation to select checkpoints and thresholds and reserve the test set for final scoring. Train each model with seeds 17, 29, and 43 and report the mean and standard deviation. Estimate a 95% confidence interval for the primary macro F1 difference by resampling test contributors in a paired bootstrap.

Classifier-only evaluation uses the 19 annotated crop classes and the top-scoring label, without rejection. End-to-end evaluation uses full images and the 18 target gesture labels. Match predicted boxes to all annotated hand boxes by descending intersection over union (IoU), at IoU ≥ 0.50, using each box at most once. A matched target with the correct emitted label is a true positive. A wrong target label counts as a false positive for the predicted class and a false negative for the true class. A missed or rejected target is a false negative. An emitted gesture on an idle hand or an unmatched box is a false positive. For the primary comparison, repeat this scoring with annotated boxes in place of the detector boxes, retaining the same checkpoint, labels, and threshold.

Table 1. Planned evaluation measures and measurement status

Metric	Definition / reporting	Measured result
Accuracy (crops)	Correct labels ÷ test crops	Not measured
Precision (per class / macro)	TP ÷ (TP + FP)	Not measured
Recall (per class / macro)	TP ÷ (TP + FN)	Not measured
F1 score (per class / macro)	Harmonic mean of precision and recall	Not measured
Hand localization recall	Share of hands matched at IoU ≥ 0.50	Not measured
Processing latency / FPS	Median and 95th percentile (ms); processed frames per second	Not measured
False activations	Events per minute of negative webcam footage	Not measured

Macro metrics give equal weight to each class. Report class counts, the crop confusion matrix, and pipeline errors by type, box size, and brightness. Test webcam operation on participants excluded from any tuning. Include prompted gestures, transitions, empty scenes, and idle hands. During negative intervals, count each new run of consecutive frames emitting the same target label as one false activation. After 100 warm-up frames, time three 60-second runs at batch size one. Measure frame delivery to label availability, excluding camera exposure and display delay.

5. Results

No original measurements are reported for the proposed system. Kapitanov et al. [4, Table 2] reported F1 scores of 97.5% for ResNet-18 and 86.4% for MobileNetV3-Small on HaGRID's 18-class full-frame classification task. Those published scores describe a different protocol. The proposed study will report the change in 18-target macro F1 when annotated boxes are replaced with automatically localized boxes, using the same CNN and rejection threshold.

6. Discussion

A large performance drop with predicted boxes would warrant inspecting missed hands and poor crops before changing the CNN. Confusions shared by both settings would point to classification errors, such as mistaking poses that differ only in finger orientation. Check whether the errors cluster in small hands, backlit scenes, or occluded poses. Compare handedness and camera angle where reliable labels are available.

A higher rejection threshold may reduce false activations while missing more valid commands. Testing on still images cannot establish behavior during gesture transitions. Evaluate any later smoothing separately because it can suppress repeated activations while adding delay.

7. Limitations and Future Work

HaGRID emphasizes prompted poses and mostly indoor settings. Its passive-hand annotations do not cover the range of natural movement and unrelated hand poses encountered in continuous video. The method also assumes visible hands and a fixed vocabulary. A useful next study would test unseen participants on a second camera, document lighting and occlusion conditions, and measure false activations in longer recordings. Dynamic gestures require a separate temporal model and a video benchmark with event annotations.

8. Conclusion

This report proposes a compact CNN for recognizing static gestures from RGB hand crops and an evaluation that separates recognition errors from hand-localization errors. HaGRID supplies an existing participant-disjoint benchmark, while a webcam test is needed to measure false activations and runtime behavior. The proposed system's accuracy, latency, and suitability for live deployment remain to be established.

Declarations

Funding: [Funding statement to be confirmed.]
Conflict of interest: [Competing interests statement to be confirmed.]
Data availability: The 18-class HaGRID release and annotations are available from the dataset authors [7].
Author contributions: [Author roles to be confirmed.]
Ethics / consent: The benchmark protocol uses an existing public dataset. Any additional participant recordings require consent and the applicable institutional review before collection.

REFERENCES

[1] W. T. Freeman and M. Roth, "Orientation Histograms for Hand Gesture Recognition," Mitsubishi Electric Research Laboratories, Tech. Rep. TR94-03, 1994. https://www.merl.com/publications/TR94-03

[2] R. F. Pinto Jr., C. D. B. Borges, A. M. A. Almeida, and I. C. Paula Jr., "Static Hand Gesture Recognition Based on Convolutional Neural Networks," Journal of Electrical and Computer Engineering, vol. 2019, Art. no. 4167890, 2019. doi:10.1155/2019/4167890.

[3] F. Zhang et al., "MediaPipe Hands: On-device Real-time Hand Tracking," arXiv:2006.10214, 2020. https://arxiv.org/abs/2006.10214

[4] A. Kapitanov, K. Kvanchiani, A. Nagaev, R. Kraynov, and A. Makhliarchuk, "HaGRID – HAnd Gesture Recognition Image Dataset," in Proc. IEEE/CVF Winter Conf. Applications of Computer Vision, 2024, pp. 4560–4569. doi:10.1109/WACV57701.2024.00451.

[5] A. Howard et al., "Searching for MobileNetV3," in Proc. IEEE/CVF Int. Conf. Computer Vision, 2019, pp. 1314–1324. doi:10.1109/ICCV.2019.00140.

[6] P. Molchanov, X. Yang, S. Gupta, K. Kim, S. Tyree, and J. Kautz, "Online Detection and Classification of Dynamic Hand Gestures with Recurrent 3D Convolutional Neural Networks," in Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2016, pp. 4207–4215. doi:10.1109/CVPR.2016.456.

[7] HaGRID project, "HAnd Gesture Recognition Image Dataset," 18-class release and annotations. https://github.com/hukenovs/hagrid/tree/Hagrid_v2

[8] Google AI Edge, "Hand landmarks detection guide." https://developers.google.com/edge/mediapipe/solutions/vision/hand_landmarker