Task-Dependent Gesture Robustness: A Detection vs. Pose Estimation Analysis

Zihao Zhan[1]

Rensselaer Polytechnic Institute, Troy NY 12180, USA zhanz@rpi.edu

Abstract. We study robustness in gesture-based HCI through a task–environment lens. Using two geometrically contrasted gestures (OpenPalm, CloseFist), three controlled test conditions—Normal Light (NL), Interference (IN), Low-Light (LL)—and four single-stage detector–pose baselines (YOLOv11n/s-pose; Roboflow 3.0 nano/small), we train and validate on NL and evaluate per split with box mAP@50 (IoU) and pose mAP@50 (OKS@0.5). We also report retained performance relative to NL to operationalize usability. Results show a major task-level divergence: detection remains near ceiling across NL/IN/LL (mean 98% with r≈ 1.0), and IN minimally affects either task, whereas pose collapses in LL (mean 60.5%, r≈ 0.64) despite strong NL/IN scores. This pattern aligns with the nature of degradations: IN behaves like additive texture that modern detectors tolerate, while LL causes information loss that erases the fine, visible cues needed for keypoints. The implication is practical: for small-vocabulary remote-control interfaces, a detection-first pipeline is robust even under LL; for high-fidelity AR/VR manipulation and ASL, reliability hinges on pose and therefore on illumination or additional sensing. Our task-centric framing and simple protocol provide a reproducible path for evaluating gesture robustness.
Github Repository: https://github.com/EdwinZhanCN/OCH-Gesture-Analysis

Keywords: Computer Vision · Gesture Recognition · Object Detection · Human-Computer Interaction · Pose Prediction.

1 Introduction

Gesture-based interaction stands as a cornerstone of Human–Computer Interaction (HCI), giving an intuitive, contact-free control across scenarios from augmented and virtual reality (AR/VR) to mobile accessibility. Recent reviews show that hand interactions are among the most widely supported techniques in mixed reality systems and products [1], [2]. Meanwhile, computer vision for hands has advanced rapidly—from real-time, on-device pipelines that decouple palm detection from hand landmarks to large-scale benchmarks for interacting hands—yielding practical models for both gesture recognition and pose estimation.

Despite this progress, a persistent gap remains between controlled laboratory results and real-world deployment. Under distribution shifts and common

corruptions (e.g., blur, noise, illumination changes), gains in image classification or stronger backbones do not automatically transfer to downstream tasks such as detection or pose estimation, motivating evaluation frameworks that speak directly to downstream use rather than upstream robustness [3], [4]. This gap is in everyday settings such as a kitchen, where hands soiled by oil or water make touchscreens impractical and contact-free gestures an immediate need, and such simple and elegant frameworks would help with application building.

We argue that robustness in vision-based gesture HCI should be framed by the **environment** $\times$ **task** $\times$ **gesture** triad. In particular, the feasibility of a closed "fist" versus an open "palm" depends critically on the underlying **task**—coarse gesture detection vs. fine-grained pose (keypoint) estimation—and on environmental stressors. Low-light conditions reduce photon counts, lower the signal-to-noise ratio, and remove high-frequency cues, thereby degrading downstream recognition; by contrast, surface interference primarily adds nuisance texture [3]. Moreover, recent work highlights the primacy of **evidence visibility** in keypoint estimation, explaining the fragility of pose when fine cues are absent [5], [6].

To put this viewpoint into practice, we introduce a task-centric evaluation framework under standardized adverse conditions, focusing on **low-light** (information loss) and **surface interference** (nuisance texture). Our contributions are: (1) a classification system distinguishing interference from data loss in gesture-based human-computer interaction. (2) a reproducible methodology featuring specific models and metrics for both detection and pose estimation; (3) empirical evidence that detection remains comparatively stable while pose estimation collapses in low-light; and (4) Advice for practitioners: shift the design focus from optimal shapes to task feasibility under specific conditions.

2 Literature Review

2.1 Application Landscape

AR/VR. By reviewing the literature on gesture recognition we found that hand interactions are among the most widely supported techniques in mixed-reality systems and products, reflecting their broad applicability across AR/VR devices. Practical pipelines typically separate palm/hand detection and hand-landmark regression (e.g., MediaPipe Hands), allowing for real-time processing on devices but relying on clear visual details for accurate control. Consequently, AR/VR applications demand precise manipulation are highly dependent on pose accuracy and sensitive to low-light information loss[7][2].

ASL recognition. Sign-language applications emphasize large vocabularies, diversity, co-articulation, and temporal dynamics. The **WLASL** benchmark (2,000 word signs, 21k videos) catalyzed deep learning approaches for SLR (Sign-Language Recognition)[8]. Moreover, recent works highlight some persistent challenges around scalability and robustness. In this domain, strong models and rich structural cues (keypoints/skeletons, phonological features) are typically indispensable in order to achieve high performance across a wide range of tasks[9].

Remote Control. By contrast, remote gesture control for smart-home/IoT scenarios targets a **small command vocabulary** and values learnability and reliability. Recent HCI studies have shown user preferences for simple, distinct gestures in domestic contexts[10]. Many such systems could easily achieve satisfactory usability with **gesture detection plus simple motion rules** (hold and swipe), without any complex skeleton or hand keypoints[11].

Taken together, these application landscapes highlight the need for **task-centric** view: **ASL and AR/VR** inherently require **pose** and therefore are limited in low light without enhanced sensory input; **remote-control** interfaces can often rely on **detection** with a compact gesture set, aligning without empirical finding that detection remains comparatively stable where pose collapses.

2.2 Models and Tasks

The COCO dataset provides broad annotations and is a common pretraining source for state-of-the-art models including the YOLO series [12], [13]. Hand gesture recognition and hand keypoint estimation decompose into two sub-tasks: bounding box/class prediction and keypoint/skeleton prediction. Box accuracy is driven by geometry/outline, whereas keypoints require fine-grained evidence (fingertips, joints). Static hand detection is typically evaluated by IoU-based box mAP, while keypoints use OKS-AP / PCK [14].

Modern pose prediction increasingly uses **single stage**, multi-task pipelines with two heads (detection and pose). The Ultralytics YOLO11 family natively supports multiple tasks, and YOLO-Pose shows end-to-end detection+keypoints with losses aligned to the **OKS** evaluation used in COCO keypoints [13]. In contrast, **two-stage** pipelines decouple palm and hand detection from landmark regression (e.g., **MediaPipe Hands**) and may be less suitable for comparative studies across gestures [7]. Because we compare OpenHand vs. CloseHand, we adopt single-stage joint models to reduce stage-induced confounds, keeping the focus on **environment** $\times$ **task** $\times$ **gesture**.

Following A. Dumitriu et al. [15], we use Roboflow for like-for-like comparisons across single-stage YOLO pose baselines. Roboflow supports keypoint projects, dataset versioning, Train 3.0 training, and hosted/edge inference; its validation/test API returns per-instance boxes plus a keypoints array (x, y, id, confidence), letting us report box mAP@50 and OKS-based pose mAP@50 side-by-side. We include Ultralytics YOLOv11-pose checkpoints and Roboflow Train 3.0 models in our comparative study [16].

2.3 Robustness and low-light condition

Robustness in real-world scenarios is primarily limited by distribution shifts and common corruptions (blur, noise, brightness, etc.). Recent studies have shown that robustness improvements achieved in upstream classification do not naturally transfer to downstream detection or dense prediction tasks; therefore, evaluation should directly target the downstream tasks themselves, rather than relying solely on upstream benchmarks. [3], [4]

Low light belongs to "information loss" degradation: a decrease in the number of photons leads to reduced SNR and the loss of high-frequency details, thereby weakening downstream identification and localization performance; related reviews summarize its conductive effects on detection/identification from both the imaging physics and enhancement methods (LLIE) perspectives[17], [18]. In contrast, surface interference is more like a texture and modern detectors are usually more resistant.

Keypoint/pose tasks are particularly sensitive to "visible evidence". Recent work emphasizes that keypoint visibility directly determines pose quality through local details (joint edges, fingertip inflection points); occlusion, blur, or low light significantly reduces keypoint similarity (OKS) and PCK metrics.[5], [19].

In conclusion, different degradations have asymmetric effects on different tasks—surface interference has a relatively small impact on detection, while low light significantly impacts pose estimation. This conclusion provides a theoretical basis for subsequent task-centric analysis of **environment** $\times$ **task** $\times$ **gesture**.

3 Methods and Data

3.1 Dataset Creation and Protocol

We collected static hand images in three environments: Normal (indoor normal lighting), Low-light (dark room, only screen backlight), Interference (under Normal lighting, the hand surface is covered with common kitchen substances, such as white sauce/baking soda). Training and validation data came from Normal, and testing was evaluated separately for the three environments to isolate the differential impact of "surface interference (additive texture)" and "low light (information loss)" on the task. We followed the approach of HaGRID by splitting the training/testing data based on subjects (subject-wise split).

Hand Gesture Classes To ensure the generalizability and reusability of the results, we deliberately selected two types of gestures with the greatest differences in geometric shape as representatives: OpenPalm (fingers fully extended and spread; large effective reflection area of the palm) and CloseFist (fingers clenched, fingertips mostly obscured; more compact finger bone contours). These two constitute a "geometric extreme pair" and underwent generalization training and comparative evaluation under a unified skeleton and the same training recipe[20].

Data Collection Environments The data collection protocol is designed to capture gesture data across the following three distinct, controlled environments:

- **Normal Environment:** This baseline condition was established indoors during daytime, with all available room lighting activated to ensure optimal

and diffuse illumination. Participants were instructed to wear dark, solid-colored clothing to maximize the contrast against the hand and minimize background interference.
- **Low-Light Environment:** To simulate conditions that induce high sensor noise, data was captured in a completely dark room. The sole source of illumination was an LCD monitor positioned in front of the participant, set to a low brightness level.
- **Interference Environment:** This condition tested robustness against visual occlusions and textural changes on the hand itself. While maintaining the lighting of the Normal Environment, participants' hands were coated with common kitchen substances. For reproducibility, specific products were used: White Sauce and ARM & HAMMER™ Pure Baking Soda.

Data Capture and Processing All images were captured using a Nikon Zfc mirrorless camera in manual mode to ensure consistency. We dynamically applied various camera adjustments to maintain a consistent image brightness under normal lighting conditions. Images were saved in the JPEG format (DX, Normal quality).

We compute an image-derived illumination proxy per split using ImageMagick: each image is converted to grayscale and we record the global mean and median (0–255) without center cropping (command pattern: `-colorspace Gray -format "%[fx:mean*255]"`). Per-image CSVs and the exact scripts, together with dataset versions, are available in our GitHub repository

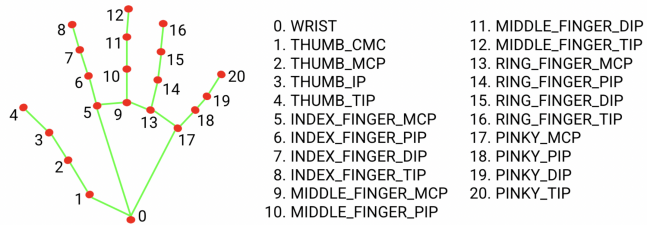
The original image resolution was 2784×1856 pixels. For compatibility with standard YOLO-based model inputs, all images were preprocessed by resizing them to a uniform 640×640 pixel resolution. Moreover, to reduce the probability of overfitting during the training process, we applied a random normal augmentation to training dataset(Hue: between -15º and +15º, Blur: Up to 2.5px, Noise: Up to 1.52% of pixels).

Dataset Composition We prepare three variants that share the same NL training/validation data but differ in the test environment:

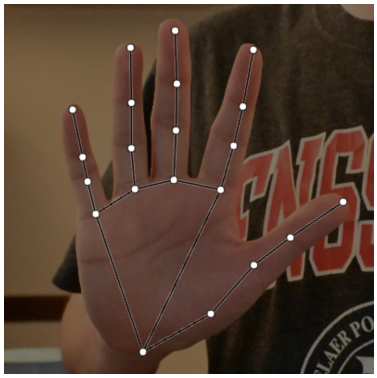
Variant	Train	Val	Test
$\mathcal{D}_{\mathrm{NL}\to\mathrm{NL}}$	NL	NL	NL
$\mathcal{D}_{\mathrm{NL}\to\mathrm{IN}}$	NL	NL	IN
$\mathcal{D}_{\mathrm{NL}\to\mathrm{LL}}$	NL	NL	LL

Table 1: Dataset variants by environment. NL: normal light; IN: normal light with interference; LL: low light. All variants use a 70/20/10 split. Overall we include 276 images; each variant's split totals 230 images (430 annotations).

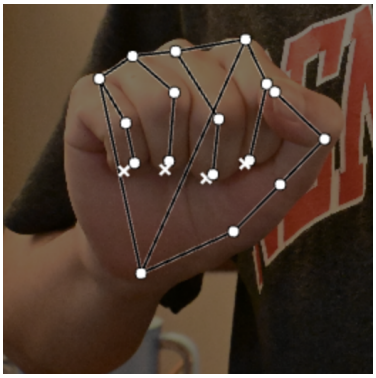
Skeleton Definition A unified 21-point skeleton, consistent with the topology used by Google's MediaPipe framework [21], was defined at the project's outset. This single skeleton was used for all gesture classes to ensure a fixed and consistent data structure for the models.



(a) The Google MediaPipe skeleton



(b) Annotation of an 'OpenPalm' gesture.



(c) Annotation of a 'CloseFist' gesture with occluded keypoints.

Fig. 1: The 21-point hand skeleton standard and annotation examples. (a) The baseline skeleton topology. (b) An example of an annotated 'OpenPalm' with all keypoints visible. (c) An example of an annotated 'CloseFist'

Labeling Procedure For each hand instance in an image, a bounding box was first drawn to encapsulate the entire hand, from the base of the wrist to the fingertips. Within this bounding box, each of the 21 keypoints was then manually placed at its correct anatomical location.

A critical aspect of our protocol was the handling of occluded keypoints, particularly for the 'CloseFist' gesture. If a keypoint was not physically visible (e.g., a fingertip tucked into the palm), it was placed at its best-estimated location on the hand's surface and explicitly marked with an "occluded" visibility flag within the annotation tool. This procedure ensures that every annotated

hand, regardless of gesture, is represented by a complete set of 21 keypoints, each with $(x, y, \text{visibility})$ coordinates, providing a rich and consistent input for model training[22].

Models and Training In training, validation and testing stages, we used four SOTA vision models provided by Roboflow platform: YOLOv11 (COCOn-Pose), YOLOv11 (COCOs-Pose), Roboflow 3.0 (COCOn-Pose) and Roboflow 3.0 (COCOs-Pose). Those lightweight models are typically used for edge detection and pose estimation tasks, which are the primary focus areas for this research. All models employ a shared-backbone, two-head design (detection for boxes/classes; pose for 21 keypoints). Pose is natively supported by YOLO11, and evaluation follows the COCO/COCO-Pose protocol with OKS similarity[13][23].

Metrics and Reporting The platform provides two metric tracks:

(i) **Detection**: box mAP@50 (IoU);
(ii) **Pose**: pose mAP@50 (OKS@0.5).

On Roboflow's dashboard, each prediction includes both `bbox` and `keypoints` for the same instance, which facilitates side-by-side comparison and CSV/JSON export. We rely on the platform's evaluators rather than re-implementing them.

4 Results

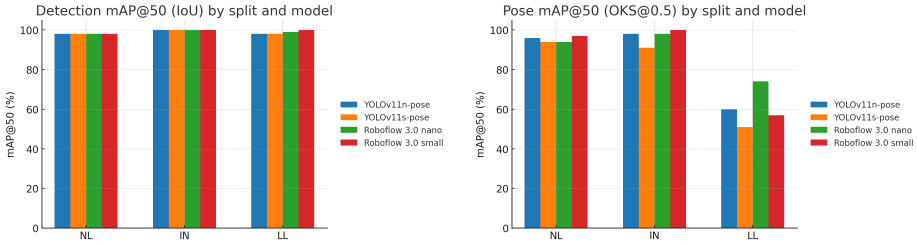
4.1 Brightness

Table 2 confirms the intended illumination separation: the **LL** split is an order of magnitude darker (MEDIAN 7.6) than **NL** (MEDIAN 103.9). The **IN** split shows substantially lower brightness than NL despite identical ambient lighting, reflecting material coverage changes on the hand (added texture rather than illumination loss). Train and validation values approximate the NL test split, indicating minimal train/validation-test brightness disparity.

Table 2: Brightness proxy per split (global grayscale mean via ImageMagick; no center crop). Higher is brighter.

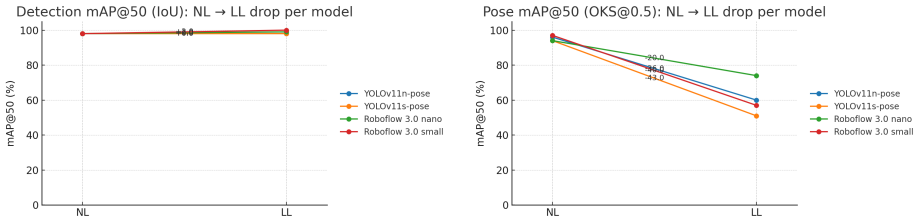
Split	AVG	MED
Train (NL, unified)	95.1	85.5
Validation (NL, unified)	97.0	104.3
Test – Normal Light (NL)	99.9	103.9
Test – Interference (IN)	63.6	63.5
Test – Low Light (LL)	7.9	7.6

4.2 Performance & Precision



(a) Detection mAP@50 (IoU). (b) Pose mAP@50 (OKS@0.5).

Fig. 2: Grouped bars by test split (NL, IN, LL) across four models.



(a) Detection NL→LL drop. (b) Pose NL→LL drop.

Fig. 3: Slopegraphs quantifying robustness from NL to LL for each model.

Fig. 2 shows per-split performance across four models. For **detection**, mAP@50 remains consistently high in all splits (NL/IN/LL). For **pose**, NL and IN are high, whereas LL exhibits a significant drop across models. Fig. 3 quantifies the NL→LL change: detection curves are flat, while pose curves show large negative slopes, with the **Roboflow 3.0 nano** remains the highest LL pose mAP among the four.

5 Evaluation

5.1 Criteria

We propose a **task-centric** criterion for evaluating the precision result, which is inspired by the work of Claudio Michaelis *et al.*[24]. **A split is usable for a given task if it satisfies both:**

(i) An absolute threshold (Detection: mAP@50 ≥ 95; Pose: mAP@50 ≥ 80)
(ii) A retention threshold $r \geq 0.8$

Where the retention threshold is introduced as defined below:

$$r = \frac{\text{mAPSplit}}{\text{mAPNL}}$$

This threshold gives a reliable performance report for a small-vocabulary gesture set. we report per-model results in Figs. 2–3 and summarize split-level means below.

- Detection mAP@50: Retention ≈ 1.01
- Pose mAP@50: Rentention ≈ 0.64

5.2 Result

By the threshold calculated and the usable definition given above, **Detection** is usable across all three test sets. While **Pose** should be seen as **unsatisfactory** under LL(Low-light) environments.

Table 3: A split is usable if it meets criteria

Split	**Det mAP@50**	**Pose mAP@50**	r_{det}	r_{pose}	**Det**	**Pose**
NL	98.00	95.25	1.00	1.00	**Yes**	**Yes**
IN	100.00	96.75	1.02	1.02	**Yes**	**Yes**
LL	98.75	60.50	1.01	0.64	**Yes**	**No**

6 Discussion

After carefully evaluating the results, we found that the gesture classes that differed in geometric shapes, **OpenPalm** and **CloseFist**, have little impact on both detection and pose accuracy across all condition and SOTA vision models. Furthermore, interference or additive texture minimally impacts detection and pose accuracy for all conditions and state-of-the-art vision models. Lastly, despite slightly lower performance in dark/lowlight conditions compared to other scenarios, the detection task still achieved very high accuracy. The only unusable

comparison pair is when model input is under LL(Low-light) condition given a pose estimation task.

These findings undoubtedly support our **task-centric** philosophy. Surface interference behaves like additive texture, to which modern vision models remain comparatively elastic. While low light causes information loss that largely hurts pose estimation, which is a task that relies on find-grained, visible cues around joints/fingertips[6]. Therefore, for remote-control interface with small command sets, a **detection-first** pipeline is usually sufficient and robust; however, **AR/VR** and **ASL** inherently depend on pose and therefore benefit more from illumination or other sensing source under low-light conditions[2].

To ground our usability rule, we normalize each split's score to the clean reference (NL) and report retained performance, where r = metrciSplit/metricNL, in line with corruption-robustness benchmarks (e.g., **ImageNet-C, COCO-C**)[25][24]. We also encourage application builders and software developers to use this metric to evaluate the usability of a specific gesture subset under the given task and environment.

7 Limitations

- **Classes.** Our study uses only two gesture classes (OpenPalm, CloseFist) to maximize geometric contrast. This keeps the analysis focused but limits coverage of richer vocabularies.
- **Illumination (no lux).** We did not perform absolute lux calibration. Instead, we use an image-derived brightness proxy (global grayscale mean/median) to confirm illumination differences across splits.
- **Static frames.** All experiments are conducted on single images; dynamic gestures and temporal cues are out of scope.

8 Conclusion and Future Work

In this work, we first investigated and review the previous work about application landscape for gesture recognition, robustness in low-light environment and the relationship between state-of-the-art vision models and its task. We clearly identified that the difference between single-staged and two-staged detectors for pose estimation such as YOLO families and MediaPipe Hand, we reviewed the trade-offs and therefore chose single-stage to reduce confounds. Furthermore, we reframe gesture robustness as a **task-environment** problem and notice the little impact of property of gesture geometry. The controlled experiment across four modern single-stage detector-pose models, we found that the detection task remains near-ceiling in Normal, Interference, and even Low-Light splits, while pose collapses in Low-Light condition, reflecting the difference between additive texture and information loss. Theses findings are consistent with robustness practices which lead us to adopt a new measurement that evaluates degraded

condition relative to a clean reference.In addition, we give prospective developers a guide to accelerate their gesture recognition applications. For small-vocabulary remote-control interfaces, a detection-first pipeline is practical and robust. For **AR/VR and ASL**, performance depends on pose and thus benefits from improved illumination or additional sensing under Low-Light conditions. The future research directions are to extend the gesture vocabularies and more continuous gestures, also to experiment with exposure-locked Low-Light environment studies for better reliability, lastly to explore cross-device evaluation to test the generality of our task-centric conclusions.

9 Acknowledgements

This research was supported by the course **CSCI-4960 Introduction to Research on Summer 2025** at Rensselaer Polytechnic Institute and its related fellows. We are grateful to **Dr. Neha Keshan** for help with idea exploration and research design, and to TA **Nipun Deelaka Pathirage** for help with dataset declaration and draft reviews, and to classmates **Momir Petrovic, Ajitesh Bankula** for help with work reviews and idea exploration.

References

[1] D. Gavgiotaki, S. Ntoa, G. Margetis, K. C. Apostolakis, and C. Stephanidis, "Gesture-based interaction for ar systems: A short review," in *Proceedings of the 16th International Conference on PErvasive Technologies Related to Assistive Environments*, Corfu, Greece: ACM, Jul. 2023, pp. 284–292. DOI: 10.1145/3594806.3594815.

[2] R. Nguyen, C. Gouin-Vallerand, and M. Amiri, "Hand interaction designs in mixed and augmented reality head mounted display: A scoping review and classification," *Frontiers in Virtual Reality*, vol. 4, Jul. 2023. DOI: 10.3389/frvir.2023.1171230.

[3] Y. Yamada and M. Otani, "Does robustness on imagenet transfer to downstream tasks?" In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA, Jun. 2022, pp. 9205–9214. DOI: 10.1109/CVPR52688.2022.00900.

[4] S. Wang, R. Veldhuis, C. Brune, and N. Strisciuglio, *A survey on the robustness of computer vision models against common corruptions*, arXiv, version dated Sept. 14, 2024, 2024. DOI: 10.48550/arXiv.2305.06024. arXiv: 2305.06024 [cs.CV].

[5] P. Sun, K. Gu, Y. Wang, L. Yang, and A. Yao, "Rethinking visibility in human pose estimation: Occluded pose reasoning via transformers," in *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, Waikoloa, HI, USA, Jan. 2024, pp. 5891–5900. DOI: 10.1109/WACV57701.2024.00580.

[6] Z. Tian *et al.*, "A survey of deep learning-based low-light image enhancement," *Sensors*, vol. 23, no. 18, p. 7763, Sep. 2023. DOI: 10.3390/s23187763.

[7] F. Zhang *et al.*, *Mediapipe hands: On-device real-time hand tracking*, 2020. DOI: 10.48550/arXiv.2006.10214. arXiv: 2006.10214 [cs.CV].

[8] D. Li, C. R. Opazo, X. Yu, and H. Li, "Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison," in *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Snowmass Village, CO, USA, Mar. 2020, pp. 1448–1458. DOI: 10.1109/WACV45572.2020.9093512.

[9] R. Rastgoo, K. Kiani, and S. Escalera, "Sign language recognition: A deep survey," *Expert Systems with Applications*, vol. 164, p. 113 794, Feb. 2021. DOI: 10.1016/j.eswa.2020.113794.

[10] M. Hosseini, H. Mueller, and S. Boll, "Controlling the rooms: How people prefer using gestures to control their smart homes," in *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems (CHI '24)*, New York, NY, USA: ACM, May 2024, pp. 1–18. DOI: 10.1145/3613904.3642687.

[11] B. I. Alabdullah *et al.*, "Smart home automation-based hand gesture recognition using feature fusion and recurrent neural network," *Sensors*, vol. 23, no. 17, p. 7523, Aug. 2023. DOI: 10.3390/s23177523.

[12] *Coco – common objects in context*, https://cocodataset.org, Accessed: Aug. 10, 2025, 2025.

[13] Ultralytics, *Yolo11 new*, https://docs.ultralytics.com/models/yolo11, Accessed: Aug. 10, 2025, 2025.

[14] G. Papandreou *et al.*, *Towards accurate multi-person pose estimation in the wild*, 2017. DOI: 10.48550/arXiv.1701.01779. arXiv: 1701.01779 [cs.CV].

[15] A. Dumitriu, F. Tăţui, A. Miron, R. T. Ionescu, and R. Timofte, "Rip current segmentation: A novel benchmark and yolov8 baseline results," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Jun. 2023, pp. 1261–1271. DOI: 10.1109/CVPRW59228.2023.00133.

[16] Roboflow, *Launch: Label, train, deploy support for keypoint detection models in roboflow*, https://blog.roboflow.com/keypoint-detection-on-roboflow, Accessed: Aug. 10, 2025, 2025.

[17] C. Li *et al.*, *Low-light image and video enhancement using deep learning: A survey*, arXiv version dated Nov. 5, 2021, 2021. DOI: 10.48550/arXiv.2104.10729. arXiv: 2104.10729 [cs.CV].

[18] S. Zheng, Y. Ma, J. Pan, C. Lu, and G. Gupta, *Low-light image and video enhancement: A comprehensive survey and beyond*, arXiv version dated Jan. 1, 2024, 2024. DOI: 10.48550/arXiv.2212.10772. arXiv: 2212.10772 [cs.CV].

[19] K. Ludwig, D. Kienzle, and R. Lienhart, "Recognition of freely selected keypoints on human limbs," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, New Orleans, LA, USA, Jun. 2022, pp. 3530–3538. DOI: 10.1109/CVPRW56347.2022.00397.

[20] A. Kapitanov, K. Kvanchiani, A. Nagaev, R. Kraynov, and A. Makhliarchuk, "HaGRID – HAnd gesture recognition image dataset," in *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, Jan. 2024, pp. 4560–4569. DOI: 10.1109/WACV57701.2024.00451.

[21] Google AI for Developers, *Hand landmarks detection guide | google ai edge*, https://ai.google.dev/edge/mediapipe/solutions/vision/hand_landmarker, Accessed: Jul. 11, 2025, 2025.

[22] Roboflow, *Annotate keypoints | roboflow docs*, https://docs.roboflow.com/annotate/annotate-keypoints, Accessed: Aug. 10, 2025, 2025.

[23] Roboflow, *Announcing roboflow train 3.0*, https://blog.roboflow.com/roboflow-train-3-0/, Accessed: Aug. 10, 2025, 2025.

[24] C. Michaelis *et al.*, *Benchmarking robustness in object detection: Autonomous driving when winter is coming*, arXiv version dated Mar. 31, 2020, 2020. DOI: 10.48550/arXiv.1907.07484. arXiv: 1907.07484 [cs.CV].

[25] D. Hendrycks and T. Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," in *International Conference on Learning Representations (ICLR)*, Mar. 2019. DOI: 10.48550/arXiv.1903.12261.