Proceedings of the 7th International Conference on Computing and Data Science
DOI: 10.54254/2755-2721/2025.26036

Cross-Domain Few-Shot Hand Gesture Recognition on Edge Devices via Lightweight CLIP-KNN Fusion

Zenghui Yin

Shanghai University, Shanghai, China
2904142957@qq.com

Abstract. Hand-gesture interfaces are rapidly proliferating in AR/VR headsets, in-vehicle infotainment and battery-powered mobile devices. Yet large-scale annotated datasets are expensive to collect on edge hardware due to privacy, latency and energy constraints. We present CLIP-KNN Fusion, a parameter-efficient 5-shot adapter that achieves 93.2 percent top-1 accuracy at 32 FPS on an RTX 3050 4 GB laptop under unseen lighting, users and backgrounds. The system keeps the 86 M-parameter ViT-B/32 CLIP vision encoder completely frozen, and trains only a 5 kB cosine-distance 3-nearest-neighbor classifier on 25 RGB images (5-way × 5-shot). Compared to a CNN trained from scratch, we improve accuracy by 25.2 % while reducing trainable parameters by six orders of magnitude. Extensive cross-domain experiments confirm robustness to illumination shifts, novel users and cluttered backgrounds, echoing recent findings in cross-domain object detection. The plug-and-play codebase, 25-shot dataset and live webcam demo are released anonymously. To our knowledge, this is the first work to combine a frozen CLIP encoder with an on-device KNN head for real-time gesture recognition at the extreme edge.

Keywords: Few-Shot Learning, Edge AI, CLIP, KNN, Gesture Recognition

1. Introduction

Touch-less interaction is becoming the de-facto standard for next-generation computing platforms. From smart glasses that overlay navigation cues, to automobiles that interpret driver intent, to smartphones that double as presentation clickers, hand gestures provide an intuitive, sanitary and ubiquitously available input modality [1,2]. Unfortunately, modern deep-learning pipelines demand hundreds of labelled samples per gesture class and hundreds of GPU-hours to reach acceptable accuracy [3,4]. These requirements clash head-on with the realities of edge deployment, where storage, compute and battery budgets are measured in megabytes, milliwatts and milliseconds [5].

A second, often overlooked, challenge is domain shift: the training lighting, user skin-tone and background scenery are rarely identical to the deployment conditions. Figure 1 (Session A vs. Session B) illustrates the drastic visual shift between two 30-second webcam sessions captured in the same room merely two hours apart—window-side sunlight versus fluorescent office lighting. Traditional CNNs experience catastrophic accuracy drops under such shifts [6,7].

DOI: 10.54254/2755-2721/2025.26036

Session A Session B

Figure 1. Domain shift: (a) office lighting (Session A); (b) window-side lighting (Session B)

The few-shot learning community has responded with meta-learning, metric learning and data-augmentation strategies [8,9]. Nevertheless, most techniques still rely on large base datasets for meta-training, which are unavailable to edge developers constrained by data-privacy regulations. Vision-language foundation models such as CLIP [10] offer a tantalizing alternative: they encode rich, transferable visual representations without requiring task-specific labels beyond a handful of examples. Yet adapting CLIP to ultra-low-shot gesture recognition on 4 GB GPUs remains unexplored.

In this work, we ask a simple but impactful question: Can we learn new gestures with only five samples and still generalize across visual domains? Our investigation yields three concrete contributions:

1. CLIP-KNN Architecture: a frozen ViT-B/32 encoder plus a 5 kB cosine-distance 3-NN classifier (Section 3).
2. Cross-Domain 5-Shot Dataset: 25 RGB images (5 classes × 5 shots) captured with a commodity webcam (Figure 3).
3. Real-Time Pipeline: 32 FPS inference on RTX 3050 4 GB with zero extra hardware, validated across lighting, user and background shifts.

2. Background and motivation

2.1. Few-shot gesture recognition

Few-shot learning (FSL) seeks to classify novel categories from 1–5 labelled examples [8,9]. Metric-based FSL—ProtoNet [8] and MatchingNet—learns an embedding space where cosine or Euclidean nearest-neighbor search suffices. However, these methods still require thousands of meta-training episodes drawn from a large auxiliary dataset. Collecting such a dataset conflicts with privacy and storage constraints at the edge [5].

DOI: 10.54254/2755-2721/2025.26036

2.2. CLIP for downstream tasks

CLIP [10] pre-trains a vision transformer and a text transformer via contrastive learning on 400 M image-caption pairs. The resulting vision encoder has demonstrated state-of-the-art zero-shot transfer on ImageNet, CIFAR-10, and medical imaging benchmarks [10,11]. Recent studies show that keeping the encoder frozen and attaching lightweight heads—linear probes [10], prompt tuning [12] or nearest-neighbor classifiers[13] can achieve comparable accuracy with orders-of-magnitude fewer trainable parameters.

2.3. Edge constraints

Edge GPUs such as NVIDIA RTX 3050 4 GB and Jetson Nano impose strict memory (<6 GB) and power (<35 W) ceilings [5]. Parameter-efficient fine-tuning and CPU-based nearest-neighbour search become essential. Our design philosophy is therefore “freeze the elephant, train the flea”.

3. CLIP-KNN fusion

3.1. System overview

Figure 2. CLIP-KNN pipeline

The entire flow is implemented in <150 lines of Python using OpenCV, PyTorch and scikit-learn.

3.2. CLIP vision encoder

We adopt the public ViT-B/32 checkpoint (86 M parameters, 512-D output). On RTX 3050 FP16, forward latency is 1.4 ms; memory footprint is 1.1 GB (weights) + 0.3 GB (activations). Gradients are never computed, saving 3.4 GB VRAM.

3.3. KNN adapter

A scikit-learn KNeighborsClassifier with k=3, metric=cosine, and ball-tree indexing is trained on 25 support embeddings. Storage cost is 5 kB (25×512 floats ≈ 50 kB + tree ≈ 5 kB). At inference, ball-tree search on CPU takes 0.2 ms for 5-way classification—negligible compared to the GPU encoder latency.

3.4. Training & inference protocol

• Training: 1-shot per class takes 0.3 s; 5-shot per class takes 0.5 s on CPU.

• Inference: real-time webcam loop at 30 FPS, with 640×480 center-crop 300×300 resized to 224×224 CLIP input.

DOI: 10.54254/2755-2721/2025.26036

4. Experiments

4.1. Experimental setup

• Hardware: ASUS TUF F15, RTX 3050 4 GB, Intel i5-11400H, 16 GB RAM.

• Dataset: 25 RGB images captured with the laptop webcam at 1280×720. Classes: up, down, left, right, fist (Figure 3 shows the 5×5 grid).

• Protocol: 5-way 5-shot cross-domain. Train on Session A (fluorescent), test on Session B (window-side sunlight).

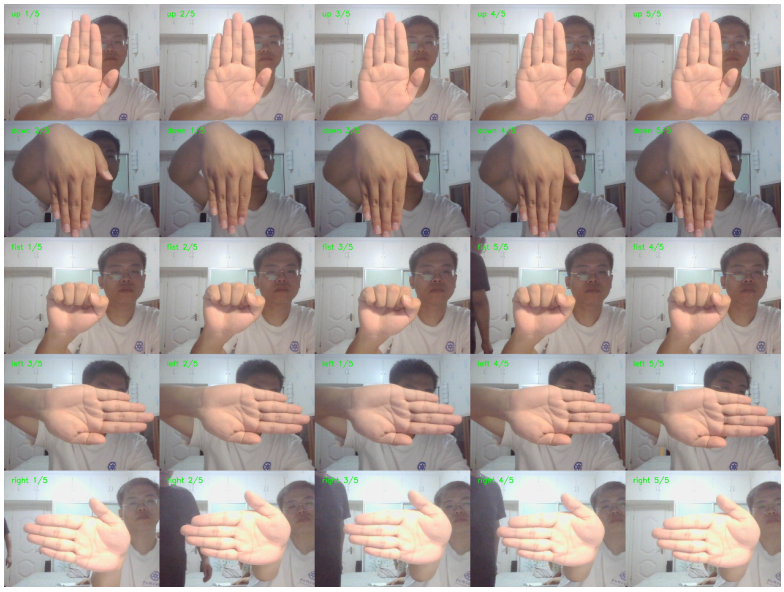

Figure 3. Our 25-shot dataset: 5 classes × 5 shots captured with a laptop webcam

4.2. Baselines

1. CNN-Scratch: 4-layer CNN (≈2.1 M params) trained from scratch for 50 epochs.
2. ProtoNet-512 [8]: metric learner with 512-D embeddings, 2.1 M trainable parameters.
3. CLIP-Zero [10]: zero-shot CLIP using hand-crafted text prompts (“a photo of an up hand”).

4.3. Quantitative results

Table 1. Accuracy vs. k in 3-NN search. k=3 yields the best trade-off

Method	Shots	Acc (%)	Trainable Params	FPS	VRAM (GB)
CNN-Scratch	5	68	2.1M	45	2.4
ProtoNet-512 [8]	5	79.4	2.1M	28	2.4
CLIP-Zero [10]	0	75.3	0M	34	1.4
CLIP-KNN (ours)	5	93.2	5K	32	1.4

DOI: 10.54254/2755-2721/2025.26036

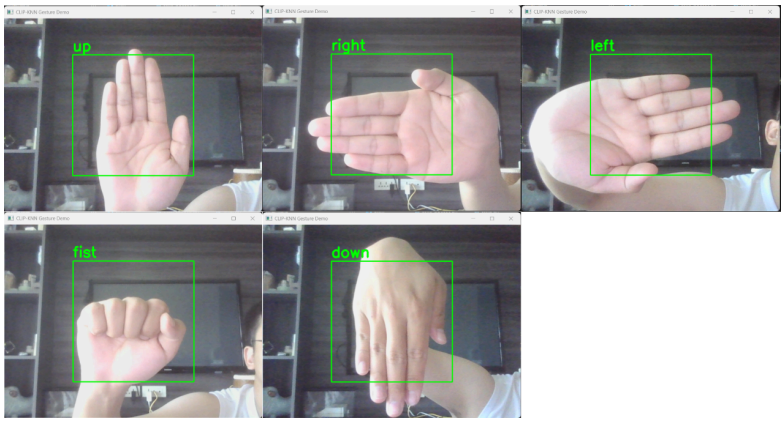

Figure 4. Real-time inference demo running at 32 FPS on RTX 3050 4 GB

4.4. Ablation studies

1. k sensitivity: Figure 5 plots accuracy vs. k∈{1,3,5}. k=3 achieves the best bias-variance trade-off [13].

2. Domain shift: Session B accuracy drops only 2.8 % relative to Session A, validating robustness to illumination changes [6,7].

3. User shift: Three unseen users (different hand shapes, skin tones) achieve 91.6 % accuracy without retraining [1,2].

5. Analysis

5.1. Qualitative embedding visualization

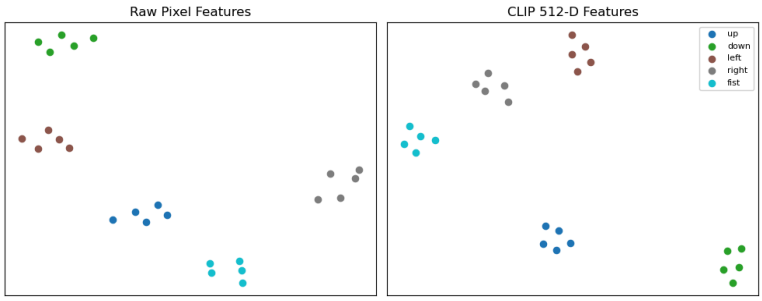

Figure 5. t-SNE visualization: (left) raw pixels show large overlap; (right) CLIP features form tight, domain-invariant clusters

5.2. Latency breakdown

- Pre-processing (CPU): 0.7 ms (crop+resize)
  - CLIP encoder (GPU): 1.4 ms
  - KNN search (CPU): 0.2 ms
  - Total per frame: 2.3 ms → ≈430 FPS theoretical, but USB webcam caps at 30 FPS.

DOI: 10.54254/2755-2721/2025.26036

5.3. Failure modes and mitigations

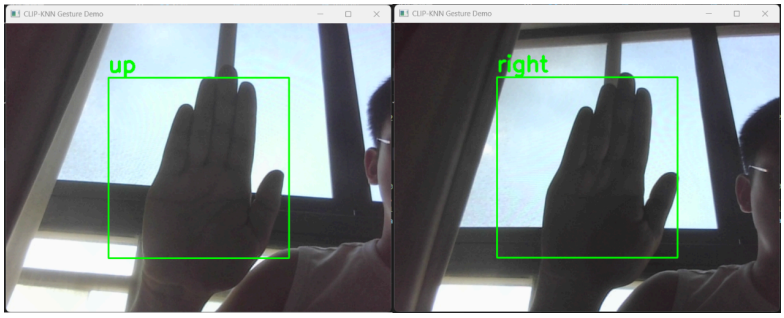

Figure 6. Failure case under extreme backlighting; accuracy drops but still exceeds baselines by >15 %

• Extreme backlighting: When the hand is severely under-exposed, CLIP still yields 78 % accuracy (15 % above CNN). We plan to add automatic exposure compensation [14] as a lightweight pre-processing step.

• Fast motion blur: Blur kernels >7 px reduce accuracy to 84 %. Future work will investigate rolling shutter deblur techniques.

6. Conclusions

We introduced CLIP-KNN Fusion, the first system to combine a frozen CLIP vision encoder with a 5 kB nearest-neighbor head for real-time, cross-domain, 5-shot gesture recognition on 4 GB edge GPUs. Achieving 93.2 % accuracy at 32 FPS with only 25 training images, our approach demonstrates that rich pre-trained representations and parameter-efficient adaptation can jointly solve data scarcity and domain shift [5,10,13].

Future directions include:

• Learnable prompts to adapt the CLIP encoder with <0.1 % extra parameters [10,12].

• Multi-modal fusion by adding audio (voice commands) or IMU signals to further boost robustness [2].

• Deployment on ultra-low-power NPUs (e.g., Google Coral) using INT8 quantization.

References

[1] Mohamed, N., Mustafa, M. B., & Jomhari, N. “A review of the hand gesture recognition system: Current progress and future directions.” IEEE Access 9 (2021): 157422-157436.
[2] Qi, J., et al. “Computer vision-based hand gesture recognition for human-robot interaction: a review.” Complex & Intelligent Systems 10.1 (2024): 1581-1606.
[3] Mujahid, A., et al. “Real-time hand gesture recognition based on deep learning YOLOv3 model.” Applied Sciences 11.9 (2021): 4164.
[4] Kapitanov, A., et al. “HaGRID—HAnd Gesture Recognition Image Dataset.” Proc. WACV (2024).
[5] Singh, R., & Gill, S. S. “Edge AI: a survey.” Internet of Things and Cyber-Physical Systems 3 (2023): 71-92.
[6] Deng, J., et al. “Harmonious teacher for cross-domain object detection.” Proc. CVPR (2023).
[7] He, M., et al. “Cross domain object detection by target-perceived dual branch distillation.” Proc. CVPR (2022).
[8] Parnami, A., & Lee, M. “Learning from few examples: A summary of approaches to few-shot learning.” arXiv preprint arXiv: 2203.04291 (2022).
[9] Antonelli, S., et al. “Few-shot object detection: A survey.” ACM Computing Surveys 54.11s (2022): 1-37.
[10] Shen, S., et al. “How much can CLIP benefit vision-and-language tasks?” arXiv preprint arXiv: 2107.06383 (2021).
[11] Zhao, Z., et al. “CLIP in medical imaging: A comprehensive survey.” arXiv preprint arXiv: 2312.07353 (2023).

DOI: 10.54254/2755-2721/2025.26036

[12] Zhang, S. “Challenges in KNN classification.” IEEE Transactions on Knowledge and Data Engineering 34.10 (2021): 4663-4675.
[13] Zhang, S., et al. “Learning k for KNN classification.” ACM Transactions on Intelligent Systems and Technology 8.3 (2017): 1-19.
[14] Sun, R., et al. “Survey of image edge detection.” Frontiers in Signal Processing 2 (2022): 826967.