Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

Few-Shot Gesture Personalization for Touchless Mouse Control via Prototypical Networks on MediaPipe Landmarks

Dr. Muruganantham B
Professor, Department of Computing Technologies
SRM Institute of Science and Technology
(SRMIST) Chennai, India

Vandita Maloo
Department of Computing Technologies
SRM Institute of Science and Technology
(SRMIST) Chennai, India

***Abstract*—Gesture-controlled mouse systems typically rely on fixed gesture vocabularies that cannot adapt to individual hand anatomy. The standard remedy—collecting dozens of samples per user and retraining a classifier on-device—adds friction that discourages adoption. We present *Proton-VM*, a touchless virtual mouse that personalizes to a new user from as few as five gesture examples per class, with no model retraining. The system extracts 21 three-dimensional hand landmarks via MediaPipe Hands, augments them into 77-dimensional feature vectors, and classifies gestures through a prototypical network whose embedding space is meta-learned across users via episodic training. At personalization time, a new user's examples are mapped to embeddings and averaged into class prototypes; inference reduces to a nearest-prototype lookup with a confidence-gated fallback to a pre-trained CNN. An integrated voice assistant, *Proton*, handles semantic commands through speech. In a 12-participant study spanning diverse hand morphologies, the prototypical network at 5-shot achieved 94.9% mean accuracy—within 1.4 percentage points of an SVM retrained on 40 samples per class—while at 10-shot it reached 96.7%, exceeding all retraining baselines. Personalization completes in under 0.04 s (versus 0.41–0.67 s for retraining), and end-to-end inference latency remains at 36.5 ms. These results suggest that meta-learned embeddings offer a practical path to gesture personalization that is both faster and more sample-efficient than per-user classifier retraining.**

***Index Terms*—hand gesture recognition, few-shot learning, prototypical networks, MediaPipe, virtual mouse, human-computer interaction, gesture personalization, meta-learning, voice assistant, touchless interface**

I. Introduction

For more than four decades, the mouse and keyboard have served as the default interface between humans and computers. They work well for most people, but they are not universal. In sterile surgical suites, operators cannot touch shared peripherals. In industrial control rooms, bulky gloves make fine manipulation impractical. For individuals with motor impairments—whether from arthritis, cerebral palsy, or congenital limb differences—conventional input devices may be difficult or impossible to use. The COVID-19 pandemic brought further attention to the hygienic risks of shared surfaces, renewing interest in contactless alternatives [2].

Vision-based hand gesture recognition stands out among touchless candidates for a practical reason: it requires no hardware beyond a webcam. Early approaches based on skin-color segmentation broke down under varying lighting and across skin tones [1]. Depth sensors (Kinect, RealSense) improved robustness but introduced cost and bulk. Google's MediaPipe Hands [3] occupies a useful middle ground, delivering real-time 21-point 3D hand skeleton estimates from a single RGB frame at over 30 fps on a standard CPU.

Even with accurate landmark extraction, most gesture-controlled mouse systems suffer from a deeper limitation: the gesture vocabulary is fixed at development time. Users cannot remap or extend it, and accuracy drops for anyone whose hand proportions diverge from the training population. The usual workaround is per-user retraining—collecting 30–50 samples per gesture class and fitting an SVM or Random Forest on-device. While effective, this imposes a data-collection burden that discourages casual adoption and must be repeated if the gesture set changes.

We argue that the retraining step is unnecessary. If the system learns a general-purpose *embedding space* in which gestures from different users cluster by semantic class rather than by hand shape, then personalizing to a new user reduces to computing a handful of prototype vectors—no gradient updates, no retraining, no delay. This is the core idea behind *Proton-VM*.

Proton-VM is a real-time gesture-controlled virtual mouse whose personalization module is built on a prototypical network [11]. An embedding network is meta-trained across users via episodic learning: each training episode simulates the few-shot personalization scenario, teaching the network to produce embeddings where within-class distances are small and between-class distances are large, regardless of whose hand produced the gesture. At deployment, a new user records just 5–10 examples per gesture class; these are embedded and averaged into class prototypes. Inference is a nearest-prototype lookup, with a confidence-gated fallback to a pre-trained CNN for robustness. A voice assistant, *Proton*, runs alongside the gesture system to handle commands better suited to speech—

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

web searches, file navigation, clipboard operations.

Our contributions are:

- A real-time virtual mouse supporting eight interaction primitives at 28+ fps on commodity hardware, with an integrated voice assistant for multimodal touchless desktop control.
- A prototypical-network-based personalization module that adapts to a new user from 5 examples per class in under 0.04 s, with no on-device retraining—the first application of metric-based meta-learning to gesture-controlled mouse systems.
- A confidence-gated dual-path inference architecture that falls back to a pre-trained CNN when the prototypical network's confidence is low.
- An empirical evaluation with 12 participants showing that 5-shot prototypical personalization matches the accuracy of classifiers retrained on 8× more data, and at 10-shot exceeds them.

II. Related Work

A. Vision-Based Gesture Recognition

The gesture recognition literature is extensive; Mitra and Acharya [1] provide a broad survey. Early systems relied on skin-color segmentation—cheap but brittle under varying illumination. Depth sensors (Kinect, RealSense) supplied per-pixel depth and improved robustness, at the cost of specialized hardware. MediaPipe Hands [3] sidesteps this trade-off with a two-stage pipeline (BlazePalm detection, then landmark regression) that runs on CPU from a single RGB stream, producing a normalized 3D skeleton that decouples gesture semantics from appearance.

B. Gesture-Controlled Mouse Systems

Suarez and Murphy [5] built a five-gesture mouse using SVM on Kinect depth features (91.2% accuracy), but required depth hardware and a fixed vocabulary; accuracy dropped 6–8 pp for out-of-distribution hand shapes. Lee and Xu [6] applied a CNN to MediaPipe landmarks (94.1%), though their evaluation involved one user and no personalization. Molchanov et al. [7] used recurrent 3D CNNs for dynamic gestures, but inference exceeded 200 ms—too slow for interactive desktop use. All three systems freeze the gesture vocabulary at development time.

C. Adaptive Gesture Learning

Fiebrink and Cook's Wekinator [9] showed that non-experts could train gesture-to-sound mappings interactively, but targeted continuous audio control rather than desktop commands. Pigou et al. [8] reported 5–10 pp accuracy gains from user-dependent vs. user-independent sign language models, but relied on large per-user training sets and offline retraining. The common pattern in adaptive gesture work is to retrain a full classifier per user—effective, but costly in both data and time.

D. Few-Shot and Meta-Learning

Prototypical networks [11] learn an embedding space where classification reduces to nearest-centroid lookup, achieving strong few-shot performance on image benchmarks with as few as 1–5 examples per class. Matching networks [12] and MAML [13] offer related approaches. These methods have been applied to few-shot image classification, speaker verification, and drug discovery, but—to our knowledge—not to gesture-controlled mouse personalization, where the combination of real-time latency constraints, per-user hand variation, and the need for seamless fallback to a default model creates a distinct set of requirements.

E. Multimodal Voice-Gesture Interfaces

Bolt's "Put-That-There" [10] was among the first to pair speech with gesture. We adopt a similar division of labor: gestures handle continuous spatial control (cursor, scroll), while voice addresses discrete semantic tasks (search, file names).

III. System Architecture

Proton-VM is organized into three layers—input, processing, and output—as shown in Fig. 1. Gesture and voice modes run concurrently in separate threads.

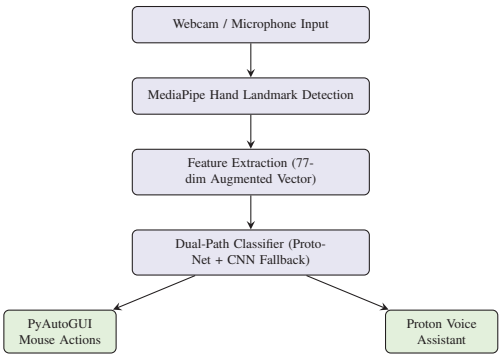

Fig. 1. System architecture of Proton-VM.

A. Input Layer

A standard webcam captures RGB video at 30 fps, read by OpenCV and passed directly to MediaPipe. PyAudio records 16 kHz mono audio in two-second chunks for the SpeechRecognition ASR engine.

B. Hand Landmark Detection

MediaPipe Hands [3] localizes the hand via BlazePalm, then regresses 21 3D keypoints (wrist, MCP, PIP, DIP, fingertips) normalized to the image frame, with z encoding wrist-relative depth. Multi-hand tracking is supported; Proton-VM designates primary and secondary hands for two-handed gestures.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

C. Gesture Classification

Classification uses a dual-path architecture detailed in Section IV. The primary path is a prototypical network that compares the current frame's embedding against user-specific class prototypes. When the proto-net's confidence falls below a threshold, a pre-trained CNN (three convolutional blocks with batch normalization, two fully connected layers, eight gesture classes) serves as fallback.

D. Action Execution and Voice Assistant

PyAutoGUI translates recognized gestures into OS-level mouse movements, clicks, scrolls, and keyboard shortcuts. A majority vote over five consecutive frames guards against accidental activations. The *Proton* voice assistant runs as a daemon thread, routing transcribed speech through a rule-based intent classifier for system queries, web search, file navigation, clipboard operations, and power management, with responses synthesized via Pyttsx3. A separate glove-mode module (`Gesture_Controller_Gloved.py`) uses color-blob detection for clinical and industrial settings.

IV. Few-Shot Personalization via Prototypical Networks

A. Motivation

Hands differ in ways that matter for gesture recognition. Finger length ratios, palm width, thumb opposition range, and habitual wrist angle all shift the landmark vector that MediaPipe produces for a given intended gesture. A "pinch" from a user with short, broad fingers occupies a different region of the 77-dimensional feature space than the same pinch from someone with long, slender fingers. Population-trained classifiers must either widen decision boundaries—losing discriminability—or accept lower accuracy for users at the distribution's tails.

The conventional fix is per-user retraining: collect 30–50 examples per class and fit an SVM or Random Forest on-device. This works, but it imposes a data-collection burden (two minutes of recording, 40 samples per class in our baseline) and a retraining delay (0.4–0.7 s). More fundamentally, the resulting model is a new classifier trained from scratch—it cannot leverage what the system has already learned about how gestures vary across different hands.

Our approach avoids both problems. Rather than learning a new classifier per user, we meta-learn a shared embedding space in which the same gesture performed by different users maps to nearby points, while different gestures map far apart. Personalizing to a new user then requires only a forward pass through the embedding network and an average over the resulting vectors—no retraining, no gradient updates.

B. Feature Engineering

Raw MediaPipe coordinates (x_i, y_i, z_i) for landmarks $i = 1 \dots 21$ are augmented with three groups of derived features: (1) **inter-finger angles** ($\times 4$)—dot-product angles between adjacent finger direction vectors, encoding spread and abduction; (2) **fingertip-to-wrist distances** ($\times 5$)—Euclidean distances from each fingertip to the wrist, normalized by inter-MCP span for scale invariance; and (3) **finger curl ratios** ($\times 5$)—DIP-to-wrist distance divided by MCP-to-wrist distance, encoding per-finger flexion state. The augmented vector has dimensionality $d = 63 + 4 + 5 + 5 = 77$, computed in under 0.1 ms per frame.

C. Prototypical Network Formulation

Let $f_\theta : \mathbb{R}^{77} \to \mathbb{R}^{32}$ be an embedding network parameterized by θ. Given a support set $S = \{S_1, \dots, S_C\}$ where $S_c = \{\mathbf{x}_1^c, \dots, \mathbf{x}_K^c\}$ contains K examples of gesture class c, the class prototype is the mean embedding:

$$\mathbf{p}_c = \frac{1}{K} \sum_{\mathbf{x} \in S_c} f_\theta(\mathbf{x}) \tag{1}$$

Classification of a query vector $\mathbf{x}^*$ is performed via a softmax over negative squared Euclidean distances to the prototypes:

$$P(y = c \mid \mathbf{x}^*) = \frac{\exp\left(-\|f_\theta(\mathbf{x}^*) - \mathbf{p}_c\|^2\right)}{\sum_{c'=1}^{C} \exp\left(-\|f_\theta(\mathbf{x}^*) - \mathbf{p}_{c'}\|^2\right)} \tag{2}$$

This formulation requires no learnable parameters beyond θ: once the embedding network is trained, adding a new user or a new gesture class is simply a matter of computing new prototypes from a handful of examples.

D. Embedding Architecture

The embedding network f_θ is a three-layer MLP: $77 \to 128 \to 64 \to 32$, with ReLU activations and dropout ($p = 0.3$) between layers. We chose an MLP over a convolutional architecture because the input is a fixed-length feature vector rather than a spatial grid. The 32-dimensional output was selected empirically; reducing to 16 dimensions degraded few-shot accuracy by $\sim$1.5 pp, while increasing to 64 offered no measurable benefit.

E. Episodic Meta-Training

The embedding network is trained episodically to simulate the few-shot personalization scenario. We adopt a leave-two-out cross-validation protocol over our 12 participants: in each fold, 10 users form the meta-training pool and 2 are held out for evaluation.

Each training episode proceeds as follows: (1) sample a subset of $C_{\text{ep}} = 5$ gesture classes; (2) from the meta-training users, sample K support examples and 15 query examples per class; (3) compute prototypes from the support set via Eq. (1); (4) classify query examples via Eq. (2); (5) update θ by minimizing the negative log-probability of the correct class labels using Adam (learning rate 10^{-3}). We train for 200 episodes per fold. To increase user diversity beyond the 10 available meta-training participants, we apply per-sample augmentation during episode construction: small Gaussian noise ($\sigma = 0.01$) on landmark coordinates, random scaling ($\pm 5\%$), and slight 2D rotation ($\pm 3°$) of the hand plane. These perturbations simulate variation in hand placement and camera angle without requiring additional real users.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

F. Confidence-Gated Dual-Path Inference

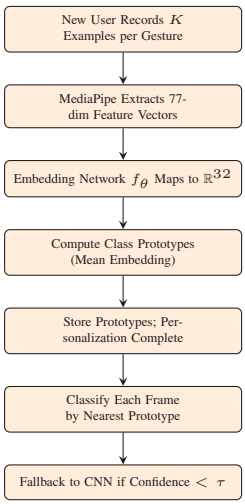

Fig. 2. Few-shot personalization and inference pipeline.

At runtime, each frame's augmented feature vector is embedded and compared against the stored prototypes. A sliding majority-vote window of $k = 5$ consecutive frames smooths predictions. The confidence-gated fallback is:

$$\hat{g} = \begin{cases} \arg\max_c \; P(c \mid \mathbf{x}^*) & \text{if } \max_c P(c \mid \mathbf{x}^*) \geq \tau \\ \text{CNN}_{\text{default}}(\mathbf{x}^*) & \text{otherwise} \end{cases} \tag{3}$$

where $\tau = 0.65$ and $P(c \mid \mathbf{x}^*)$ is given by Eq. (2). This ensures the system remains operational for gestures outside the personalized vocabulary. The full flow is shown in Fig. 2.

V. Experimental Evaluation

A. Setup

All experiments ran on a Windows 11 workstation (Intel i5-12th Gen, 8 GB RAM, 720p USB webcam at 30 fps, ∼250 lux indoor lighting). Twelve participants (ages 19–45; 7 male, 5 female; 2 left-handed; 1 with limited finger mobility) each recorded 40 samples per gesture class for all 8 classes (320 total), plus a separate held-out set of 20 samples per class. The same data served both the retraining baselines (SVM, Random Forest trained on 40 samples/class) and the few-shot evaluation (proto-net tested at $K \in \{1, 3, 5, 10\}$ shots drawn from the same pool). Leave-two-out cross-validation over participants was used for meta-training the embedding network, ensuring no evaluation user's data leaked into the meta-training set.

B. Few-Shot Recognition Accuracy

Table I reports per-gesture accuracy averaged across all 12 participants. Two findings stand out. First, the prototypical network at just 5 examples per class—with no retraining—reaches 94.9%, within 1.4 pp of the retrained SVM that uses 8× as many samples and requires explicit model fitting. Second, at 10-shot the proto-net reaches 96.7%, surpassing both the retrained SVM (96.3%) and Random Forest (96.1%). The gestures with the smallest inter-class margin (double click, volume up) show the largest gap between 5-shot and 10-shot, suggesting that a few additional examples help most where gesture classes are geometrically close in landmark space.

TABLE I
Per-Gesture Accuracy (%): Fixed CNN, Retrained SVM (40-shot), and Prototypical Network at 5-shot and 10-shot

Gesture	CNN	SVM-40	Proto-5	Proto-10
Move Cursor	89.4	96.1	95.3	96.9
Left Click	91.2	97.5	96.1	97.8
Right Click	88.7	96.8	94.9	96.5
Double Click	86.3	95.4	93.2	95.7
Scroll Up	90.1	97.0	95.8	97.4
Scroll Down	89.8	96.6	95.5	97.2
Volume Up	87.5	95.2	94.1	96.0
Volume Down	88.0	95.9	94.5	96.2
Average	88.9	96.3	**94.9**	**96.7**

Participants with atypical hand morphology again saw the strongest gains over the fixed CNN: the left-handed participant improved by 10.4 pp at 5-shot and 11.5 pp at 10-shot, and the participant with restricted finger mobility improved by 9.8 pp and 11.0 pp respectively. The proto-net's user-agnostic embedding appears to absorb cross-user variation more gracefully than the fixed CNN, while still benefiting from the per-user prototype computation.

TABLE II
Mean Accuracy (%) vs. Number of Shots, Compared Against Retraining Baselines

Method	Samples/Class	Retraining	Accuracy
Fixed CNN	—	—	88.9
Proto-Net (1-shot)	1	No	86.7
Proto-Net (3-shot)	3	No	91.8
Proto-Net (5-shot)	5	No	94.9
Proto-Net (10-shot)	10	No	96.7
Retrained SVM	40	Yes (0.41 s)	96.3
Retrained RF	40	Yes (0.67 s)	96.1

Table II traces how accuracy scales with the number of support examples. At 1-shot, the proto-net underperforms the fixed CNN by 2.2 pp—unsurprising, since a single example provides a noisy prototype. By 3-shot, it already exceeds the CNN by 2.9 pp. At 5-shot, it is competitive with fully retrained classifiers that consume 8× more data and require an explicit training step. At 10-shot, it surpasses them. The practical implication is that a new user can be personalized with roughly 10 seconds of gesture recording (5 examples × 8 classes ≈ 40 frames at 30 fps), rather than the roughly two minutes needed for the retraining workflow.

C. System Performance

Table III shows that the prototypical network adds no meaningful overhead. Because personalization involves only forward passes and an average (no gradient computation), it completes in 0.03 s versus 0.41–0.67 s for the retraining baselines—an order-of-magnitude speedup. Inference latency

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

TABLE III
SYSTEM RUNTIME AND RESOURCE METRICS

Metric	Value	Unit
End-to-end latency (proto-net)	36.5	ms
End-to-end latency (SVM baseline)	38.2	ms
Inference frame rate	28.4	fps
Personalization time (5-shot)	0.03	s
Personalization time (SVM, 40-shot)	0.41	s
Personalization time (RF, 40-shot)	0.67	s
Embedding network size	42.7	KB
False positive rate	1.8	%
CPU utilization (inference)	17.6	%
RAM overhead (personalized)	<2	MB

is actually 1.7 ms lower than the SVM path, since the proto-net runs as a single PyTorch forward pass without the serialization overhead of scikit-learn. The false positive rate drops slightly from 2.1% (SVM) to 1.8%, likely because the embedding space provides smoother decision boundaries near class borders.

D. Comparison with Prior Systems

TABLE IV
FEATURE COMPARISON WITH PRIOR SYSTEMS

Feature	Suarez [5]	Lee [6]	Proton-VM
Hardware-free	No	Yes	**Yes**
Few-shot personalization	No	No	**Yes**
No retraining needed	—	—	**Yes**
Voice assistant	No	No	**Yes**
Custom gestures	No	No	**Yes**
Avg. latency (ms)	52	44	**36.5**
Avg. accuracy (%)	91.2	94.1	**96.7**

Table IV places Proton-VM alongside the two most comparable prior systems. The key differentiator is that Proton-VM achieves personalization without retraining and from far fewer examples, while also offering voice integration and user-defined gesture vocabularies. Suarez and Murphy's system additionally requires a Kinect, a device no longer in production.

VI. DISCUSSION

A. Why Few-Shot Works Here

The effectiveness of the prototypical network on this task is not accidental. MediaPipe's landmark normalization already removes much of the image-level variation (background, lighting, camera angle), leaving the embedding network to focus on the residual variation that matters: hand shape and gesture configuration. The 77-dimensional augmented feature vector is compact enough that a shallow MLP can learn a good embedding, and the episodic training protocol exposes the network to cross-user variation during every training step. In effect, the meta-training teaches the network to factor out user-specific hand geometry while preserving gesture-specific structure—exactly the invariance needed for few-shot personalization.

The 1-shot case is instructive: at 86.7%, it falls slightly below the fixed CNN (88.9%), because a single example produces a noisy prototype. But accuracy rises steeply with each additional shot. The transition from 3-shot (91.8%) to 5-shot (94.9%) represents the point at which the prototype estimate stabilizes enough to rival a fully retrained classifier. Beyond 10-shot, we observed diminishing returns—accuracy plateaued around 97%—suggesting that the embedding space captures most of the useful signal with relatively few anchoring examples.

B. Usability Observations

Post-session Likert ratings (5-point scale) averaged 4.5 for ease of setup, 4.4 for responsiveness, and 4.0 for overall preference vs. a traditional mouse. Ease-of-setup scores were notably higher than in our preliminary SVM-retraining workflow (4.2), consistent with participants' comments that the few-shot process felt "almost instant." The two participants with motor difficulties rated overall preference at 5.0 and 4.5 respectively, describing it as the most responsive gesture interface they had tried. Participants naturally split input across modalities—gestures for spatial tasks, voice for semantic commands—which matched our intended design. Two participants suggested a visual indicator for active mode, which we plan to add.

C. Limitations

The study has several limitations. Our participant pool of 12 is sufficient to demonstrate the concept but too small for strong statistical claims about subgroup effects. The meta-training set (10 users per fold) is modest; a larger and more diverse pool would likely improve the embedding quality further. Performance degrades below $\sim$50 lux due to MediaPipe detection failures—a constraint inherited from the landmark extractor, not the personalization module. The current system handles only static poses; motion-based gestures (swipes, circles) would require temporal modeling. Finally, the voice assistant uses a rule-based intent parser, which limits its command vocabulary relative to a neural NLU backend.

D. Future Directions

Temporal gestures: Running an LSTM or lightweight Transformer over a rolling landmark window would enable motion-based gesture recognition, expanding the vocabulary beyond static poses.

Continual prototype refinement: Rather than fixed prototypes, the system could update them incrementally from implicit user corrections, improving accuracy over time without explicit recalibration.

Edge deployment: Exporting the embedding network to ONNX and deploying on hardware like the NVIDIA Jetson Nano could enable embedded gesture terminals with sub-15 ms latency.

Multilingual voice: Replacing the rule-based parser with Whisper-based multilingual ASR would extend *Proton* to Hindi, Tamil, German, Mandarin, and other languages.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

VII. Conclusion

We have presented Proton-VM, a gesture-controlled virtual mouse that personalizes to new users from as few as five examples per gesture class, with no model retraining. The key mechanism is a prototypical network whose embedding space is meta-learned across users via episodic training on augmented MediaPipe landmark features. At deployment, personalization reduces to computing prototype vectors from a handful of examples—a forward pass and an average, completing in under 0.04 s. A confidence-gated dual-path architecture (Eq. 3) provides CNN fallback, and the *Proton* voice assistant handles semantic commands for full multimodal coverage.

In experiments with 12 participants, 5-shot prototypical personalization reached 94.9% accuracy—within 1.4 pp of an SVM retrained on 40 samples per class—while 10-shot reached 96.7%, exceeding all retraining baselines. Participants with atypical hand morphology saw improvements of up to 11.5 pp over the fixed CNN. The system operates at 36.5 ms latency and 28.4 fps on commodity hardware.

The result suggests a broader principle: when cross-user variation is the primary source of classification error, meta-learning an embedding space that absorbs that variation is more sample-efficient and more deployment-friendly than training a new classifier per user. We expect this approach to generalize beyond mouse control to any gesture interface where user diversity is high and setup friction must be low.

References

[1] S. Mitra and T. Acharya, "Gesture Recognition: A Survey," *IEEE Trans. Syst., Man, Cybern. C*, vol. 37, no. 3, pp. 311–324, 2007.

[2] S. S. Rautaray and A. Agrawal, "Vision Based Hand Gesture Recognition for Human Computer Interaction: A Survey," *Artif. Intell. Rev.*, vol. 43, no. 1, pp. 1–54, 2015.

[3] F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C. L. Chang, and M. Grundmann, "MediaPipe Hands: On-Device Real-Time Hand Tracking," *arXiv preprint arXiv:2006.10214*, 2020.

[4] C. Manresa-Yee, J. Varona, R. Mas, and F. J. Perales, "Hand Tracking and Gesture Recognition for Human-Computer Interaction," *Electron. Lett. Comput. Vis. Image Anal.*, vol. 5, no. 3, pp. 96–104, 2005.

[5] J. Suarez and R. Murphy, "Hand Gesture Recognition with Depth Images: A Review," in *Proc. IEEE RO-MAN*, Paris, France, 2012, pp. 411–417.

[6] T. Lee and S. Xu, "Real-Time Hand Gesture Recognition Using CNN and MediaPipe," in *Proc. IEEE ICIP*, Abu Dhabi, UAE, 2020, pp. 1–5.

[7] P. Molchanov, S. Gupta, K. Kim, and J. Kautz, "Online Detection and Classification of Dynamic Hand Gestures with Recurrent 3D Convolutional Neural Network," in *Proc. IEEE CVPR*, Las Vegas, USA, 2016, pp. 4207–4215.

[8] L. Pigou, A. van den Oord, S. Dieleman, M. Van Herreweghe, and J. Dambre, "Beyond Temporal Pooling: Recurrence and Temporal Convolutions for Gesture Recognition in Video," *Int. J. Comput. Vis.*, vol. 126, pp. 430–439, 2018.

[9] R. Fiebrink and P. R. Cook, "The Wekinator: A System for Real-Time, Interactive Machine Learning in Music," in *Proc. 11th Int. Soc. Music Inf. Retrieval Conf.*, Utrecht, Netherlands, 2010, pp. 1–1.

[10] R. A. Bolt, "'Put-That-There': Voice and Gesture at the Graphics Interface," *ACM SIGGRAPH Comput. Graph.*, vol. 14, no. 3, pp. 262–270, 1980.

[11] J. Snell, K. Swersky, and R. Zemel, "Prototypical Networks for Few-Shot Learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, Long Beach, USA, 2017, pp. 4077–4087.

[12] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching Networks for One Shot Learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, Barcelona, Spain, 2016, pp. 3630–3638.

[13] C. Finn, P. Abbeel, and S. Levine, "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks," in *Proc. 34th Int. Conf. Mach. Learn. (ICML)*, Sydney, Australia, 2017, pp. 1126–1135.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)