Revista de Informática Teórica e Aplicada - RITA - ISSN 2175-2745
Vol. 33, Num. 2 (2026) 44-51

RESEARCH ARTICLE

Human Gesture Recognition using CNNs for Human-Robot Interaction

Reconhecimento de Gestos Humanos com CNNs para Interação Humano-Robô

Werikson Frederiko de Oliveira Alves[1]*, Michel Melo da Silva[1], Alexandre Santos Brandão[2]

Abstract: This paper proposes a deep learning approach for human gesture recognition to support real-time teleoperation of mobile robots in outdoor environments. A lightweight ResNet18 architecture is adapted via transfer learning, combining a personalized dataset captured with a RealSense camera and a reorganized subset of HMDB. A modular pipeline was developed and evaluated under nine experimental configurations, considering different optimizers, datasets, and backbone freezing levels. Results demonstrate that models trained with combined data and adaptive fine-tuning strategies achieve high accuracy and strong generalization under varied lighting and background conditions. The best configuration reached 97.9% test accuracy, reinforcing the potential of CNNs for robust gesture-based human–robot interaction.
Keywords: gesture recognition — convolutional neural networks — transfer learning — human-robot interaction

Resumo: Este artigo propõe uma abordagem de aprendizado profundo para o reconhecimento de gestos humanos, a fim de dar suporte à teleoperação em tempo real de robôs móveis em ambientes externos. Uma arquitetura ResNet18 leve é adaptada por meio de aprendizado por transferência, combinando um conjunto de dados personalizado capturado com uma câmera RealSense e um subconjunto reorganizado do HMDB. Um pipeline modular foi desenvolvido e avaliado em nove configurações experimentais, considerando diferentes otimizadores, conjuntos de dados e níveis de congelamento de backbone. Os resultados demonstram que os modelos treinados com dados combinados e estratégias de ajuste fino adaptativas alcançam alta precisão e forte generalização sob condições variadas de iluminação e fundo. A melhor configuração atingiu 97,9% de precisão no teste, reforçando o potencial das CNNs para uma interação robusta entre humanos e robôs baseada em gestos.
Palavras-Chave: reconhecimento de gestos — redes neurais convolucionais — aprendizado por transferência — interação humano-robô

[1] *Departamento de Informática, Universidade Federal de Viçosa (UFV), Brazil*
[2] *Departmento de Engenharia Elétrica, Universidade Federal de Viçosa (UFV), Brazil*
***Corresponding author**: werikson.alves@ufv.br
DOI: http://dx.doi.org/10.22456/2175-2745.150950 • **Received:** 15/10/2025 • **Accepted:** 03/11/2025

1. Introduction

Hand gesture recognition has emerged as a promising modality for intuitive and contactless Human–Robot Interaction (HRI), particularly in tasks involving mobile robotics and Unmanned Aerial Vehicle (UAV) teleoperation. Gesture-based interfaces provide fast and natural communication mechanisms, eliminating the need for physical controllers and enhancing usability in dynamic or unstructured environments. This approach has found applications in infrastructure inspection [1], precision agriculture [2], public surveillance [3], and search-and-rescue operations [4].

Despite their potential, traditional gesture recognition systems based on handcrafted features—such as contour descriptors, motion trajectories, or background subtraction—often fail under real-world conditions involving lighting variation, occlusions, and background clutter [5, 6, 7]. Moreover, these methods typically generalize poorly to unseen users or environments due to their strong reliance on predefined features.

In recent years, Convolutional Neural Networks (CNNs) have significantly advanced the state of the art by enabling automatic learning of robust and hierarchical visual representations from raw data [8]. Architectures such as ResNet [9] combine depth and efficiency, and when paired with transfer learning strategies, they allow for effective adaptation to domain-specific tasks even with limited annotated data [10, 11].

Another critical component of gesture recognition systems is the data acquisition process. Various sensors—such as

the Kinect [12] and Intel RealSense [13]—have been widely used to collect RGB-D data, while motion capture systems (MoCap) have been employed for precise 3D tracking of human body posture [14]. Although these modalities enhance robustness, challenges such as class imbalance, gesture ambiguity, and acquisition variability still limit generalization in real-world scenarios.

Recent studies have explored lightweight CNN-based classifiers trained on specific datasets, sometimes employing domain adaptation or fine-tuning techniques to improve cross-environment performance [15, 16]. However, many of these systems remain limited either by their dependence on controlled indoor settings or by the lack of diversity in the training data.

To address these challenges, this paper proposes a modular and efficient gesture recognition system for tasks involving UAV teleoperation, leveraging a ResNet18-based CNN adjusted through transfer learning. The system classifies five predefined gestures associated with semantic commands such as *takeoff*, *land*, and *inspect*. In contrast to previous approaches that rely on a single source of gestures or a fixed environment, our method integrates generic and customized datasets, aiming at cross-domain generalization.

Furthermore, the proposed system is particularly suited for real-world scenarios that demand intuitive interaction under uncertain conditions. Representative applications include aerial inspection of infrastructure in areas where manual control is impractical, rapid deployment in search-and-rescue missions where traditional controllers may be infeasible, and hands-free operation in agriculture or surveillance tasks where the operator must remain mobile and attentive.

2. Related Work

Gesture recognition has been a longstanding research topic in the computer vision community. Early approaches relied on manual feature engineering, using optical flow, silhouette contours, or skeletal tracking to classify gestures. As discussed by Szeliski [5], such techniques can be effective in constrained scenarios but often fail to generalize under real-world variability such as lighting changes, occlusions, or background clutter.

The introduction of deep learning brought substantial improvements. Goodfellow et al. [8] demonstrated how CNNs could learn spatial hierarchies of features directly from raw image data, eliminating the need for handcrafted descriptors. More recent studies, such as those by Shahin et al. [11] and Zhang et al. [10], have shown that transfer learning significantly boosts performance when training data is scarce. By fine-tuning pretrained architectures like VGG, ResNet, or MobileNet, these models achieve competitive results even on small, task-specific datasets.

To further improve recognition accuracy, several authors have explored the use of multimodal data. Carvalho et al. [12], for instance, utilized the Kinect v2 sensor to capture both RGB and depth data, while Jeeru et al. [13] employed the Intel RealSense D435 for RGB-D acquisition of dynamic gestures. These works highlight the benefits of incorporating depth information for better motion representation.

Other studies have emphasized the importance of dataset diversity and class balance in promoting generalization. Reddy et al. [17] and Ayesha et al. [18] analyzed the impact of imbalanced gesture distributions and proposed methods for improving learning robustness across gesture categories.

In the specific context of Human–Robot Interaction (HRI), Patronas et al. [4] proposed a dedicated gesture vocabulary designed for UAV command and control. Alves et al. [14] adopted a motion capture system (MoCap) to track gesture execution in high-precision indoors environments, demonstrating the feasibility of non-contact gesture-driven control. However, these systems often depend on specialized hardware or operate under constrained indoor conditions.

Building on these contributions, the present work adopts a deep learning approach based on ResNet18, leveraging transfer learning to combine generic and personalized gesture datasets. Unlike previous studies that rely on fixed data sources or high-cost sensors, this work emphasizes generalization across acquisition domains by evaluating multiple training strategies—including freezing configurations and optimizer choice—to support gesture recognition for UAV teleoperation in real-world environments.

3. Methodology

This work proposes a deep learning-based approach for recognizing human gestures in outdoor scenarios, with the aim of enabling vision-based teleoperation of mobile robots.

3.1 Model Architecture

The network architecture is based on ResNet18, a residual convolutional neural network with moderate depth and suitable for real-time inference on resource-constrained hardware. Transfer learning was employed by initializing the model with ImageNet-pretrained weights. The final fully connected layer was replaced with a new linear layer consisting of five output units, corresponding to the five gesture classes.

To mitigate overfitting and control the level of fine-tuning, the initial layers of the ResNet18 backbone were frozen during training. Specifically, one to three residual blocks closest to the input were excluded from gradient updates, depending on the experiment. This strategy is based on the assumption that lower layers capture generic visual features (e.g., edges and textures) that are transferable across domains, whereas deeper layers are more task-specific [19]. Freezing early blocks allows the model to retain general-purpose representations while adapting higher-level features to the gesture recognition task.

3.2 Action Classes Definition

The gesture recognition system was designed to classify five action classes commonly adopted in the literature for com-

manding unmanned aerial vehicles (UAVs) and other robotic platforms [16, 20, 21].

Although similar gesture semantics have been explored in previous studies, the present dataset was newly collected to meet the specific requirements of this work. The acquisition process differs from earlier implementations in terms of sensor configuration, environment, and annotation protocol, thus requiring a dedicated gesture dataset.

Each gesture was constrained to a four-second time window, allowing for quick execution while ensuring the system remains responsive to subsequent commands. Figure 1 illustrates the gesture classes included in the dataset:

1. **Inspect (I):** Raise the left hand, forming a 90° angle between the forearm and arm, while moving the hand to the left.
2. **Land (L):** Raise the left hand to approximately face height.
3. **Follow me (F):** Bring the right hand up to the left shoulder.
4. **Take-off (T):** Raise the right hand to approximately face height.
5. **Photography (P):** Raise the right hand, forming a 90° angle between the forearm and arm, while moving the hand to the right.

3.3 Dataset Description and Preparation

This study uses two datasets—one generic and one personalized—along with a third configuration combining both:

- **GEN:** A generic dataset derived from the HMDB dataset [22]. We selected video frames corresponding to five gesture-like action categories: Follow me (F), Inspect (I), Land (L), Photography (P), and Take-off (T). Each frame was extracted at 640×480 resolution from RGB-only videos and manually curated to isolate static gesture instances. The resulting dataset comprises 1,044 images across varied lighting conditions, backgrounds, and subjects. The distribution is imbalanced, with classes ranging from 86 to 293 samples (see Table 1).
- **PER:** A personalized dataset collected using an Intel RealSense D435 camera. All images are RGB frames captured under real-world, controlled indoor. Each gesture was performed by a single operator and captured with fixed camera parameters at a resolution of 640×480 pixels. The dataset includes 1,375 images, equally distributed among the five gesture classes (275 images per class).
- **GEN+PER:** A merged dataset obtained by concatenating the GEN and PER datasets to increase visual diversity in terms of backgrounds, lighting, and performers. This configuration aims to evaluate cross-domain generalization.

Table 1 summarizes the per-class sample counts for each dataset.

Table 1. Sample distribution per class in each dataset.

Class	F	I	L	P	T	Total
GEN	204	293	224	86	237	1,044
PER	275	275	275	275	275	1,375
GEN+PER	479	568	499	361	512	2,419

All images were stored in directories compatible with the format from the `torchvision.datasets` module. Following dataset loading, the images were resized to 224×224 pixels and normalized using ImageNet mean and standard deviation.

To improve generalization and reduce overfitting, the training set was augmented using random transformations:

- Hue shift: ±15 units
- Saturation variation: ±25 units
- Brightness variation: ±15 units
- Exposure adjustment: ±10 units
- Gaussian blur: up to 2.5 pixels
- Random noise: up to 0.1% of image pixels

A stratified splitting strategy was employed to ensure that each class was proportionally represented across the training (74%), validation (11%), and testing (15%) subsets. This approach helps maintain class balance across all phases of training and evaluation.

Furthermore, the full training pipeline, source code, pretrained models, and experimental outputs are publicly available in the project repository[1]. However, the complete datasets include human subjects and are therefore not publicly released; representative samples and collection details can be shared upon request.

3.4 Experimental Setup and Implementation Details

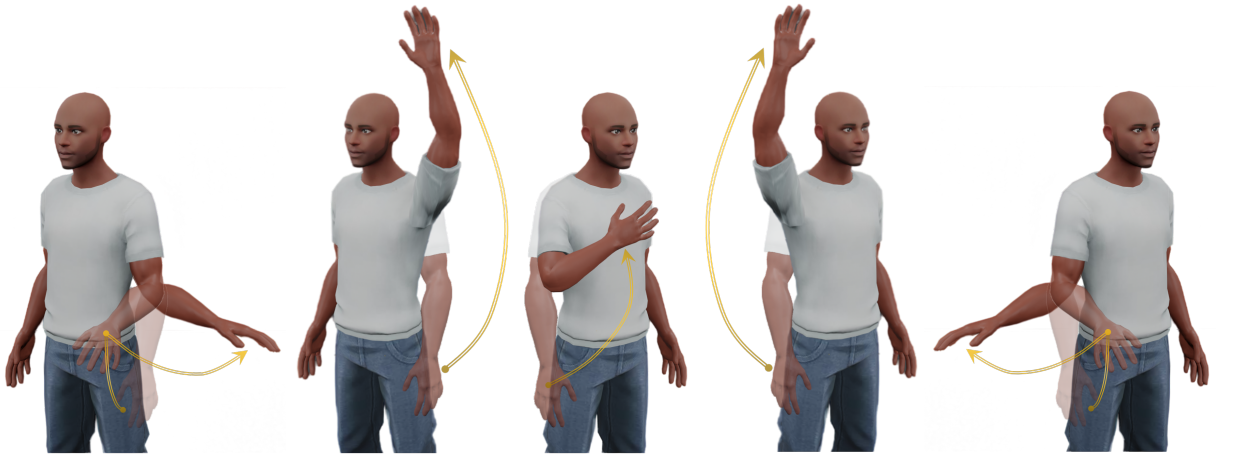
To investigate the effect of transfer learning under various conditions, nine experimental modes were defined. These combine different pretraining sources (GEN, PER, or ImageNet), optimizers, and backbone freezing strategies.

All experiments were implemented in PyTorch using a modular codebase. The training and evaluation pipeline was organized into separate modules responsible for model definition, training, validation, and result logging.

The experiments were conducted on a personal computer equipped with an Intel Core i5 processor, 32GB of RAM, and an NVIDIA RTX 2050 GPU with 4GB of VRAM.

Each model was trained for 50 epochs using a batch size of 32 and a fixed learning rate of 1×10^{-5}. Both `SGD` and

[1] ⟨https://github.com/WeriksonAlves/transfer_gesture_cnn⟩



a) Inspect (I) *b*) Land (L) *c*) Follow me (F) *d*) Take-off (T) e) Photography (P)

Figure 1. Gesture classes with corresponding movement definitions.

`Adam` optimizers were evaluated. A `StepLR` scheduler was applied every 20 epochs with a decay factor of $\gamma = 0.5$. The loss function was `CrossEntropyLoss`, and classification accuracy was used as the primary evaluation metric. Table 2 summarizes the training configuration.

Table 2. Training hyperparameters used in all experiments.

Parameter	Value
Epochs	50
Batch size	32
Learning rate	1×10^{-5}
Optimizers	`SGD`, `Adam`
Scheduler	`StepLR` (step=20, $\gamma = 0.5$)
Loss function	`CrossEntropyLoss`
Evaluation metric	Accuracy (%)
Frozen layers	1–3 initial residual blocks

Nine training modes were evaluated, varying in terms of pretraining source (ImageNet or GEN), dataset composition (GEN, PER, or GEN+PER), optimizer, and the number of frozen residual blocks at the beginning of the ResNet18 backbone. These modes are summarized in Table 3, which also includes the validation and test accuracy obtained for each configuration.

To evaluate model performance, we computed overall classification accuracy as well as precision, recall, and F1-score (both macro and weighted). We also report qualitative results, including training curves, confusion matrices, and per-class accuracy breakdowns, to better understand the behavior of each configuration under varying data domains.

4. Results

This section presents the results of gesture classification experiments using ResNet18 under different training regimes, including fine-tuning from pretrained models (ImageNet or GEN) and training from scratch. The goal is to analyze how transfer learning, optimizer choice, and data composition affect model performance, particularly in cross-domain scenarios relevant to UAV teleoperation.

4.1 Validation Performance

Each model was trained and validated using the dataset splits defined in Section 3.4. Table 3 summarizes the best validation accuracy for each configuration, along with the optimizer used, the number of frozen residual blocks, and the corresponding test accuracy on the combined GEN+PER dataset—which is used as a proxy for cross-domain generalization.

Figure 2 illustrates the evolution of validation accuracy across training epochs for all configurations. Notably, configuration ID 2 (trained on the PER dataset using SGD) achieved 100% validation accuracy but only 68.52% on the GEN+PER test set. This highlights a clear overfitting to the training domain. On the other hand, configuration ID 3—trained directly on GEN+PER with Adam and minimal freezing—showed stable and high accuracy across both validation (96.64%) and test (97.91%), suggesting better generalization to heterogeneous input distributions.

4.2 Cross-Domain Generalization

To evaluate generalization, all models were tested on the unified GEN+PER dataset, which combines gesture instances from different domains and acquisition conditions. As shown in Table 3, models trained on isolated datasets (IDs 1, 2, 7 and 8) showed poor transferability—despite high validation scores—demonstrating a lack of robustness when exposed to unseen gesture styles, lighting, or backgrounds.

Configurations trained directly on GEN+PER (IDs 3, 4, 5, and 6) showed significantly better generalization. Model ID 3 achieved the best result, with 97.91% test accuracy. This improvement is attributed to two main factors: (i) training on heterogeneous data allows the model to learn domain-invariant features, and (ii) the use of Adam with shallow freezing (1

Table 3. Training configurations and resulting performance on validation and test sets.

ID	Description	Frozen Blocks	Optimizer	Step	Relative (min)	Validated Dataset	Validation Acc. (%)	Test Acc. (%)
0	From Scratch → GEN+PER	All	SGD	50	23.81	GEN+PER	75.89	75.49
1	FT: ImageNet → GEN	3	SGD	50	7.35	GEN	97.22	57.10
2	FT: ImageNet → PER	3	SGD	24	4.43	PER	100.00	68.52
3	TL: GEN → GEN+PER	1	Adam	50	13.54	GEN+PER	96.64	97.91
4	TL: GEN → PER	1	Adam	50	7.79	PER	95.17	87.60
5	TL: ImageNet → GEN+PER	1	Adam	50	13.72	GEN+PER	91.70	92.48
6	TL: ImageNet → GEN+PER	1	SGD	50	13.57	GEN+PER	85.38	87.05
7	TL: ImageNet → PER	1	Adam	50	8.31	PER	90.69	65.04
8	TL: ImageNet → PER	1	SGD	50	7.74	PER	95.17	63.51

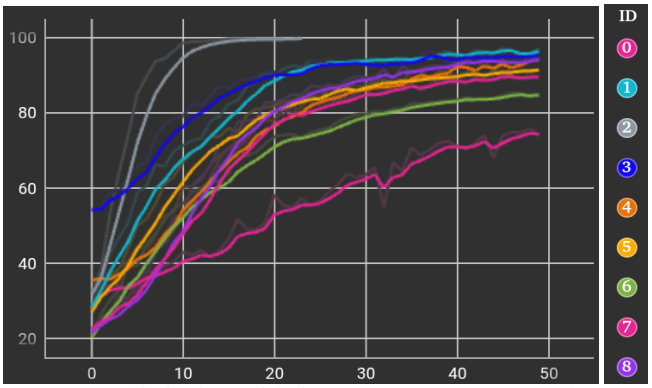


Figure 2. This is the validation accuracy curve over the training epochs, showing the time spent training each model.

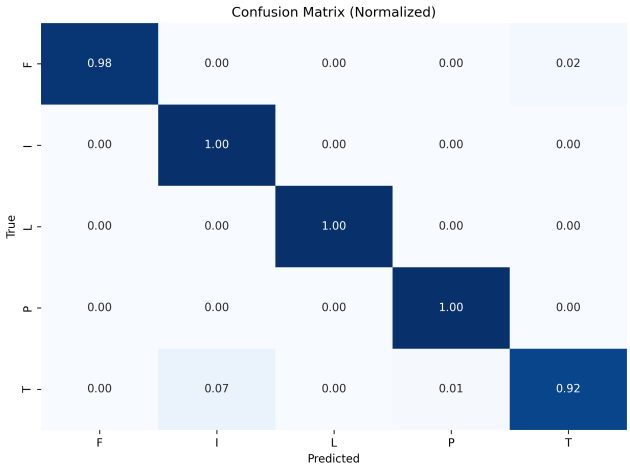


Figure 3. Normalized confusion matrix for model ID 3 (GEN → GEN+PER, Adam).

residual block frozen) promotes flexible adaptation of the final layers while preserving stable low-level representations.

4.3 Per-Class Performance and Error Analysis

To further analyze classification behavior beyond aggregate accuracy, a detailed per-class evaluation was performed using model ID 3—the configuration that achieved the highest test accuracy (97.91%, see Table 3). This model was pretrained on GEN and fine-tuned on the combined GEN+PER dataset using the Adam optimizer.

Figure 3 presents the normalized confusion matrix. The model exhibited high precision and recall for most gesture classes, particularly for *I*, *L*, and *P*, with F1-scores above 0.95 (Table 4). However, higher confusion was observed between visually similar gestures, such as *F* and *T*, and, to a lesser extent, between *T* and *I*. These cases indicate lower recall for class *F* and slightly reduced precision for class *T*, suggesting that spatial overlap and execution variability pose challenges for the CNN model.

These findings emphasize the importance of going beyond global accuracy by incorporating class-level metrics. In future work, we plan to include occlusion-based robustness evaluations and a deeper error breakdown to better characterize the model's failure modes and guide the design of disambiguation mechanisms in gesture vocabularies.

Table 4. Classification Report for model ID 3, which performed better in both stages.

	precision	recall	f1-score	support
F	1.00	0.98	0.99	142
I	0.94	1.00	0.97	166
L	1.00	1.00	1.00	149
P	0.98	1.00	0.99	108
T	0.98	0.92	0.95	153
accuracy			0.98	718

4.4 Inference Strategy and Temporal Considerations

Each recorded gesture sequence spans four seconds and was captured at 30 fps, resulting in about 120 frames per instance. However, the current model was trained and evaluated using static RGB images individually extracted from these sequences, without temporal aggregation. In practice, this results in a frame-by-frame classification strategy, where each image is assigned an independent label.

While suitable for offline benchmarking, this inference approach may lead to unstable predictions in real-time scenarios, particularly in dynamic outdoor environments where gesture execution may vary slightly across frames. For example, rapid

hand transitions or occlusions can cause abrupt shifts in the model's output, potentially resulting in conflicting or unsafe commands being sent to the UAV.

To mitigate this issue, future versions of the system will implement temporal decision mechanisms, such as majority voting or soft aggregation over a sliding time window. This will enable the model to produce a single, stable output per gesture sequence, enhancing safety, consistency, and reliability during continuous deployment. Such strategies are critical for robust Human–Robot Interaction (HRI) in real-world teleoperation tasks.

4.5 Discussion and Insights

The experimental results provide key insights into the role of pretraining, data diversity, and fine-tuning strategies for CNN-based gesture recognition.

First, we performed an additional experiment in which a ResNet18 model was trained entirely from scratch (without any pretrained weights). This configuration reached a maximum validation accuracy of 75.89% after 50 epochs, with an absolute improvement of 46.47 percentage points from the starting accuracy of 27.87%. While this represents a significant learning curve, its final performance was still lower than that of pretrained models (e.g., 97.22% for ImageNet $\rightarrow$ GEN, and 96.64% for GEN $\rightarrow$ GEN+PER). These results empirically support the common assumption that transfer learning accelerates convergence and improves final accuracy in small- to medium-sized datasets [10]. The training curves shown in Figure 2 also reflect the slower convergence and lower saturation level of the **from scratch** model compared to pretrained configurations.

Second, we observed that models trained with mixed-domain data (GEN+PER) consistently outperformed single-domain models across both test accuracy and class balance. For instance, models trained only on GEN or PER reached high validation accuracy within-domain but showed significant accuracy drops when tested on GEN+PER (e.g., from 100% to 68.52% in ID 2). This reinforces the importance of dataset diversity for robust generalization across gesture performers, lighting conditions, and backgrounds.

Finally, fine-tuning strategies also proved critical. The best-performing configurations (e.g., ID 3 and ID 5) used the Adam optimizer combined with shallow freezing (only one residual block frozen), enabling efficient adaptation while preserving low-level visual features learned during pretraining. In contrast, deeper freezing or the use of SGD generally resulted in lower test performance and slower convergence.

Taken together, these findings indicate that effective gesture recognition benefits from three aligned factors: (i) transfer learning to accelerate learning and improve generalization, (ii) mixed-domain training data to enhance robustness, and (iii) fine-tuning protocols that balance stability and flexibility. Table 3 summarizes the configurations, and their impact is visualized in Figure 2.

5. Concluding Remarks

This study proposed a deep learning approach for gesture recognition using CNNs, aiming to enhance teleoperation capabilities in mobile robotic platforms, particularly in outdoor and unstructured environments. The motivation stems from the shortcomings of traditional computer vision methods, which are often sensitive to lighting changes, occlusions, and variations in camera perspective.

We adopted a transfer learning strategy using the ResNet18 architecture and systematically evaluated nine training configurations that varied in dataset composition, optimizer, and feature freezing levels. The modular pipeline enabled automated training, reproducible evaluation, and consistent comparison across experiments.

The experimental results support the following key conclusions:

- **Transfer learning from ImageNet** is effective even in small, domain-specific gesture datasets, accelerating training and improving initial convergence.
- **Cross-domain generalization** is significantly enhanced by dataset diversity. Models trained on combined data (GEN+PER) consistently outperformed single-source configurations.
- **Domain adaptation strategies**, such as pretraining on GEN followed by fine-tuning on PER, improved performance over direct fine-tuning on personalized data alone.

These findings confirm the effectiveness of combining deep CNNs with transfer learning to build gesture recognition systems capable of generalizing to heterogeneous, real-world scenarios—an essential requirement in Human–Robot Interaction (HRI).

However, the current study has some limitations that should be addressed in future work:

- **Frame-wise inference only:** The system was trained on static images extracted from video sequences, and classification was performed on individual frames without temporal aggregation. This may lead to inconsistent predictions in real-time applications.
- **Limited subject diversity:** The personalized dataset was collected from a single operator, which may restrict the model's ability to generalize to different users or motion styles.
- **No training from scratch:** All configurations relied on pretrained weights from ImageNet; the contribution of pretraining was not isolated through experiments without transfer learning.

Future work will address these issues by:

- Expanding the dataset with more subjects, gesture styles, and environmental conditions to improve robustness and fairness.
- Integrating temporal smoothing or voting mechanisms to produce more stable predictions during gesture execution.
- Exploring the use of pose-based representations (e.g., YOLOv8-Pose) to enhance interpretability and reduce sensitivity to visual noise.
- Deploying and validating the system in real-time robotic control frameworks using middleware platforms such as ROS 2.

In practical terms, the proposed approach opens new opportunities for deploying vision-based control interfaces in critical field applications. UAV-based inspections of bridges, power lines, and remote facilities can particularly benefit from hands-free, intuitive command execution. Similarly, in emergency scenarios—such as disaster response or agricultural monitoring—the ability to issue high-level commands via gesture reduces both cognitive and physical workload, enabling safer and more responsive robotic interaction.

Moreover, addressing the limitation of frame-wise inference identified in this work will be essential for future deployments. Integrating temporal decision mechanisms—such as majority voting or sliding-window smoothing—across frames within a fixed time interval can produce more deterministic and stable gesture classification. This improvement will strengthen the system's reliability in real-world UAV teleoperation and other time-critical Human–Robot Interaction (HRI) scenarios.

In summary, this work contributes evidence that combining generic and personalized data in transfer learning pipelines can improve the reliability of vision-based interaction systems under variable conditions.

Acknowledgements

The authors gratefully acknowledge the financial support provided by the Brazilian research agencies CNPq, CAPES and FAPEMIG. We are also grateful to Instituto de Inteligência Artificial e Computacional (Idata-UFV), FINEP (IA-AD-UFV #0284/22), and Cluster/UFV for providing GPU resources.

Author Contributions

The first author was responsible for the design, implementation, and execution of the proposed system, as well as the analysis and interpretation of the results. The other two authors provided guidance, supervision, and critical review throughout the development of the project and preparation of the manuscript.

References

[1] MOYSIADIS, V. et al. An integrated real-time hand gesture recognition framework for human–robot interaction in agriculture. *Applied Sciences*, MDPI, v. 12, n. 16, p. 8160, 2022.

[2] LIU, C.; SZIRÁNYI, T. Gesture recognition for uav-based rescue operation based on deep learning. In: *Improve*. [S.l.: s.n.], 2021. p. 180–187.

[3] GUNASUNDARI, S. et al. Gesture controlled drone swarm system for violence detection using machine learning for women safety. In: SPRINGER. *Int. Conf. on Intelligent Sustainable Systems*. [S.l.], 2023. p. 221–236.

[4] PATRONA, F.; MADEMLIS, I.; PITAS, I. An overview of hand gesture languages for autonomous uav handling. *2021 Aerial Robotic Systems Physically Interacting with the Environment*, IEEE, p. 1–7, 2021.

[5] SZELISKI, R. *Computer vision: algorithms and applications*. [S.l.]: Springer Nature, 2022.

[6] BASILIO, V. T.; CARVALHO, K.; BRANDAO, A. Reconhecimento de açoes por rna em aplicaçoes de robotica social. *XIV Simpósio Brasileiro de Automaçao Inteligente, Ouro Preto/MG*, 2019.

[7] RAY, P.; REDDY, S. S.; BANERJEE, T. Various dimension reduction techniques for high dimensional data analysis: a review. *Artificial Intelligence Review*, Springer, v. 54, n. 5, p. 3473–3515, 2021.

[8] GOODFELLOW, I. et al. *Deep learning*. [S.l.]: MIT press Cambridge, 2016. v. 1.

[9] HE, K. et al. Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2016.

[10] ZHANG, A. et al. *Dive into deep learning*. [S.l.]: Cambridge University Press, 2023.

[11] POURBAHRAMI, S. et al. A survey of neighborhood construction algorithms for clustering and classifying data points. *Computer Science Review*, v. 38, p. 100315, 2020. ISSN 1574-0137.

[12] CARVALHO, K. B. de; BASÍLIO, V. T.; BRANDÃO, A. S. Action recognition for educational proposals applying concepts of social assistive robotics. *Cognitive Systems Research*, Elsevier, v. 71, p. 1–8, 2022.

[13] JEERU, S. et al. Depth camera based dataset of hand gestures. *Data in Brief*, v. 45, p. 108659, 2022. ISSN 2352-3409.

[14] ALVES, W. F. d. O.; MOREIRA, K. S.; BRANDAO, A. S. Interaçao com o ambiente mediante classificaçao por distância euclidiana. In: *Congresso Brasileiro de Automática*. [S.l.: s.n.], 2022. v. 3, n. 1.

[15] SHEENA, C. V.; NARAYANAN, N. K. Static gesture classification and recognition using hog feature parameters and k-nn and svm-based machine learning algorithms. In: BAJPAI, M. K.; SINGH, K. K.; GIAKOS, G. (Ed.). *Machine Vision and Augmented Intelligence—Theory and Applications*. Singapore: Springer Singapore, 2021. p. 157–166. ISBN 978-981-16-5078-9.

[16] CARVALHO, K. B. de et al. Gestures-teleoperation of a heterogeneous multi-robot system. *The Int. Journal of Advanced Manufacturing Technology*, Springer, v. 118, n. 5, p. 1999–2015, 2022.

[17] REDDY, G. T. et al. Analysis of dimensionality reduction techniques on big data. *IEEE Access*, v. 8, p. 54776–54788, 2020.

[18] AYESHA, S.; HANIF, M. K.; TALIB, R. Overview and comparative study of dimensionality reduction techniques for high dimensional data. *Information Fusion*, v. 59, p. 44–58, 2020. ISSN 1566-2535.

[19] YOSINSKI, J. et al. How transferable are features in deep neural networks? *Advances in neural information processing systems*, v. 27, 2014.

[20] HUDSON, T. M. et al. Gesture and emotion recognition system for autonomous vehicle control. In: *2025 Brazilian Conference on Robotics (CROS)*. [S.l.: s.n.], 2025. v. 1, p. 1–5.

[21] ALVES, W. F. d. O.; BARCELOS, C. O.; BRANDAO, A. S. A gesture recognition system for robot guidance applications. In: SOCIEDADE BRASILEIRA DE AUTOMÁTICA (SBA). *Congresso Brasileiro de Automática (CBA)*. 2024. p. 1–6. Disponível em: ⟨https://www.sba.org.br/cba2024/papers/paper_3971.pdf⟩.

[22] KUEHNE, H. et al. HMDB: a large video database for human motion recognition. In: *Proceedings of the International Conference on Computer Vision (ICCV)*. [S.l.: s.n.], 2011.