© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715462

An Open-Set Framework for Robust Hand Gesture Classification

PROF. B. VENKATESWARLU[1], M. SUSMITHA[2], R. VARALAKSHMI[3], S K. MUNEER[4], T. PAVAN[5]

[1, 2, 3, 4, 5]*Department of Information Technology, RVR & JC College of Engineering, Chowdavaram Guntur.*

Abstract- ***Hand gesture recognition has become an important component in many modern applications such as human–computer interaction, virtual environments, and sign language interpretation. Most existing gesture recognition systems operate under a closed-set assumption, where the gesture categories used during testing are the same as those present in the training stage. However, this assumption is often unrealistic in real-world environments, where systems may encounter new gesture types or variations that were not previously observed. To overcome this limitation, the concept of open-set hand gesture recognition has gained increasing attention. In an open-set scenario, a recognition system should be capable of identifying known gesture classes while also handling unfamiliar gestures that appear during deployment. This paradigm enables models to adapt to new gesture categories with limited examples and improves their ability to function in dynamic and unconstrained environments. By focusing on the open-set learning framework, this work aims to make gesture recognition systems more flexible, scalable, and suitable for practical real-world applications where the set of possible gestures cannot always be predefined.***

Keywords— Hand Gesture Recognition, Open-Set Learning, Viewpoint Variation, Joint-based Features, Incremental Learning

I. INTRODUCTION

Hand gestures play a vital role in human non-verbal communication, as they allow people to convey meaningful information quickly and naturally [1]. With the rapid development of sensing technologies and depth cameras, hand gesture recognition has attracted significant attention in computer vision research [2], [3]. Gesture recognition systems are widely used in applications such as sign language interpretation [4], [5], human–computer interaction, and virtual or augmented reality environments [6]. These applications require recognition systems that can accurately interpret hand movements under diverse real-world conditions.

Recent advances in deep learning have significantly improved the performance of hand gesture recognition systems. Convolutional neural networks and other deep learning architectures have achieved high accuracy on benchmark datasets by automatically extracting discriminative features from visual data [7], [8]. Despite these improvements, many existing approaches are designed under a closed-set assumption, where the gesture classes encountered during testing are assumed to be the same as those used during training. In practical scenarios, however, this assumption is often unrealistic. Real-world environments may contain new gesture variations, unseen hand shapes, or different camera viewpoints that were not present in the training data.

To address these limitations, the concept of open-set hand gesture recognition has been introduced. In an open-set setting, the recognition system must be capable of identifying known gestures while also handling unfamiliar gestures or variations that appear during deployment. Compared with traditional closed-set recognition systems, open-set recognition provides greater flexibility and adaptability, making it more suitable for real-world gesture-based applications.

One of the major challenges in gesture recognition arises from the complex structure of the human hand. The human hand contains multiple joints and possesses a high degree of freedom, which leads to large variations in gesture appearance across different viewpoints [19]. Furthermore, self-occlusion between fingers can make gesture interpretation more difficult when images are captured from unconstrained viewing angles [20], [21]. As a result, methods that rely solely on 2D image features often struggle to generalize across different viewpoints and hand configurations.

To improve robustness, several studies have explored the use of hand joint information for gesture recognition. Because hand gestures are closely related to the spatial arrangement of finger joints, incorporating joint position information can provide valuable structural cues for recognition tasks [19], [22], [23]. However, accurate estimation of hand joint positions remains a challenging problem due to occlusion, viewpoint changes, and variations in hand shapes. Even advanced hand pose estimation approaches may produce inaccurate joint predictions in complex scenarios [20], [24], which can negatively affect gesture classification performance. Therefore, it is important to develop methods that effectively combine image-based features and joint-based representations to reduce the influence of viewpoint variations and improve gesture recognition reliability. In this work, we investigate the problem of open-set hand gesture recognition and propose an approach that focuses on extracting view-independent features while incorporating structural information from hand joints. By integrating visual features with joint-based representations, the proposed framework aims to improve gesture recognition performance in unconstrained environments where viewpoint changes and hand shape variations are common.

II. RELATED WORK

Vision-based hand gesture recognition can generally be divided into two main categories: static gesture recognition and dynamic gesture recognition [30]. Static gesture recognition focuses on identifying gestures from a single image frame, whereas dynamic gesture recognition analyzes a sequence of frames to recognize gestures that involve motion over time [31]. In this work, the focus is placed on static hand gesture recognition, as it plays an important role in many real-time interaction systems.

Early research in hand gesture recognition mainly relied on hand-crafted feature extraction techniques. Methods such as SIFT descriptors [33], image moments [34], and Gabor filters [35] were commonly used to capture visual characteristics of hand shapes and gestures. These approaches typically combined feature descriptors with traditional classifiers to perform gesture recognition [32]. Although such methods achieved reasonable performance in controlled environments, their effectiveness was often limited when applied to different datasets or real-world conditions due to their dependence on manually designed features [8]. In recent years, deep learning approaches have significantly improved the performance of hand gesture recognition systems. Convolutional neural networks (CNNs) and other data-driven models can automatically learn discriminative features directly from images, eliminating the need for manual feature design. For example, Li et al. [36] proposed an end-to-end CNN-based framework for hand gesture recognition, while Tan et al. [8] introduced an enhanced densely connected convolutional network to improve feature propagation and gradient flow. These deep learning methods have achieved high recognition accuracy on several benchmark datasets [7], [8]. However, most of these approaches operate under a closed-set assumption, where the gesture classes used during testing are the same as those used during training. This assumption restricts the applicability of these models in real-world environments, where gestures may appear in different forms, viewpoints, or hand configurations.

To improve the robustness of gesture recognition systems, several studies have explored the use of hand joint information. Since hand gestures are closely related to the spatial arrangement of finger joints, incorporating joint-based features can provide useful structural information for gesture classification [37], [38]. Some methods directly utilize estimated joint positions to recognize gestures. For example, Sharma et al. [39] used 3D hand joint coordinates as input to a gesture recognition network, while Li et al. [22] proposed a spatial fuzzy matching approach that compares query gestures with reference gestures based on joint positions. Similarly, Alam et al. [23] developed a gesture recognition method that establishes matching relationships using visible fingertip combinations.

Although joint-based approaches can provide valuable structural information, they strongly depend on the accuracy of hand pose estimation. Estimating precise joint locations is a challenging task due to factors such as occlusion, viewpoint variations, and the high degrees of freedom in the human hand [19], [20]. Inaccurate joint estimation may significantly affect gesture classification performance. Furthermore, gestures captured from different viewpoints may appear substantially

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715462

different, even when representing the same gesture [21]. These challenges highlight the need for gesture recognition methods that can effectively integrate both image-based features and joint-based representations to improve robustness under varying viewpoints and hand configurations.

III. PROPOSED METHOD

A. Problem Formulation

Traditional hand gesture recognition systems typically operate under a closed-set assumption, where the gesture categories available during testing are identical to those used during training. Let the training dataset be defined as

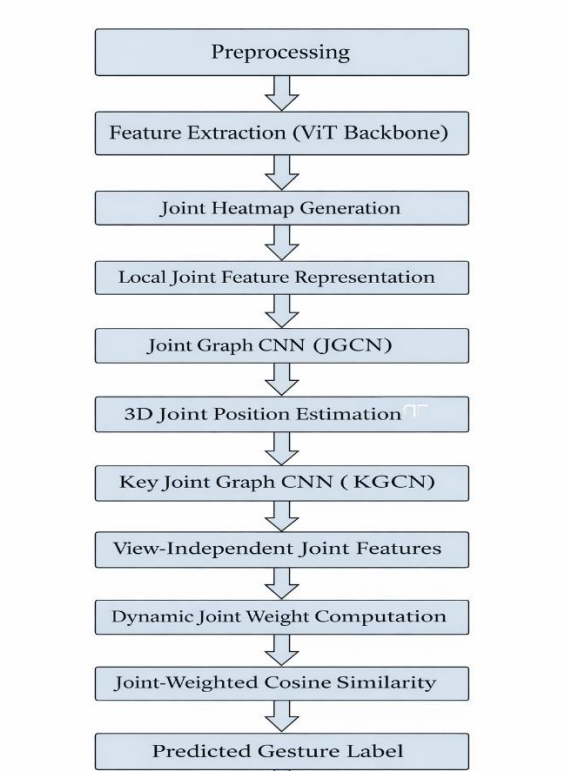
$$D = \{(x_i, y_i)\}_{i=1}^{n}$$

where $x_i$denotes an input gesture image and $y_i \in Y$represents the corresponding gesture label. The set Ycontains all predefined gesture classes used during training.

However, real-world environments often contain gesture variations and unknown patterns that are not present in the training data. Therefore, this work focuses on the open-set hand gesture recognition problem, where the recognition model must correctly classify known gestures while remaining robust to variations caused by viewpoint changes, hand shapes, and occlusions.

Given an input image x, the goal of the model Θis to extract discriminative gesture representations and predict the corresponding gesture label $y$from the set of known gesture classes. To improve recognition performance under challenging viewing conditions, the proposed framework integrates both image-based features and joint-based structural information.

3.1. Block Diagram



3.2. Algorithm 1: Feature Extraction (FE)

Input:

Depth image $x \in \mathbb{R}^{H \times W}$

Output:

Local joint-aware feature representation f_l

Steps:

1.Input Image Acquisition

Receive the depth image $x$with spatial resolution $H \times W$.

2.Image Patch Generation

Divide the input image into patches of size $P \times P$. Compute the number of patches:

$$N = \frac{HW}{P^2}$$

3.Transformer-Based Feature Extraction

Feed the image patches into the Vision Transformer (ViT-Base) network.

4.Embedding Generation

Process the patches through multiple Transformer layers to obtain embedding vectors of dimension D.

5.Feature Map Construction

Reshape the embedding vectors to form the feature map:

$$f_x \in \mathbb{R}^{D \times \sqrt{N} \times \sqrt{N}}$$

6.Joint Heatmap Generation

Pass the intermediate embedding vectors through a Deconvolution Network to generate the joint heatmap:

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715462

$$M \in \mathbb{R}^{|J| \times H_m \times W_m}$$

where $|J|$ represents the number of hand joints.

7.Heatmap Resizing

Resize the heatmap to match the spatial resolution of the feature map:

$$M' \in \mathbb{R}^{|J| \times \sqrt{N} \times \sqrt{N}}$$

8.Joint Feature Sampling

Flatten the heatmap and feature map representations.

9.Joint-Aware Feature Extraction

Compute the joint-aware feature representation:

$$f_l = M' f_x^T$$

10.Output

Return the joint-aware feature matrix:

$$f_l \in \mathbb{R}^{|J| \times D}$$

Algorithm 2: Viewpoint Influence Elimination (VIE)

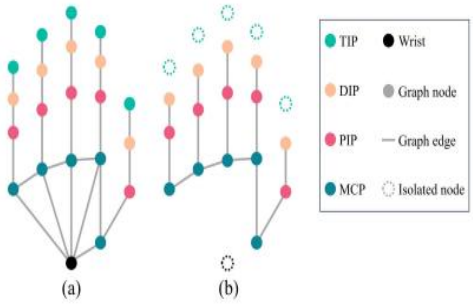


Fig. 1. The graph structure of (a) joint graph convolutional network, (b) key joint graph convolutional network.

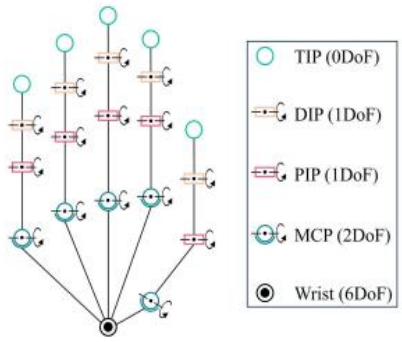


Fig. 2. The hand kinematic chain model

Input:

Local joint feature representation f_l

Output:

View-independent joint feature representation f_{vi}

Steps

1. Graph Construction

Represent the hand joints as nodes in a graph structure.

Define edges between neighbouring joints to capture spatial relationships among joints.

2. Joint Feature Propagation using JGCN

Feed the local joint features f_l into the Joint Graph Convolutional Network (JGCN).

$$f_{vd} = \Theta_{JGCN}(f_l)$$

where f_{vd} denotes the view-dependent joint features.

3. 3D Joint Position Estimation

Estimate the 3D hand joint coordinates using a position decoder:

$$\hat{P}^{3D} = \Theta_P(f_{vd})$$

Compute the regression loss to supervise the learning process:

$$\min_{\theta_{JGCN}, \theta_P} \|\Theta_P(f_{vd}) - P^{3D}\|^2$$

where P^{3D} represents the ground-truth 3D joint positions.

4. Selection of Key Joints

Identify key joints that determine hand gesture configuration:

- Metacarpophalangeal joints (MCP)
- Proximal interphalangeal joints (PIP)
- Distal interphalangeal joints (DIP)

These joints are used to model the structural properties of gestures.

5. View-Independent Feature Extraction

Feed the view-dependent features f_{vd} into the Key Joint Graph Convolutional Network (KGCN):

$$f_{vi} = \Theta_{KGCN}(f_{vd})$$

where f_{vi} denotes the view-independent joint feature representation.

6. Canonical Joint Position Prediction

Predict joint angles:

$$\alpha_K = \Theta_A(f_{vi})$$

Compute canonical joint positions using the hand kinematic model:

$$\hat{P}_c = \Phi(\alpha_K \mid l)$$

8. Canonical Joint Loss

Minimize the difference between predicted and ground-truth canonical joint positions:

$$\min_{\theta_{KGCN}, \theta_A} \|\Phi(\alpha_K \mid l) - P_c\|^2$$

where P_c represents the ground-truth canonical joint positions.

9. Output

Return the final view-independent joint features f_{vi} which are used for gesture classification.

Algorithm 3: Joint-Weighted Classification (JWC)

Input:

- View-independent joint feature vectors $f_{vi} = \{f_{vi,i}\}_{i\in K}$
- Prototype features of each gesture class $w_y = \{w_{y,i}\}_{i\in K}$
- Canonical joint positions of query gesture p_i
- Canonical joint positions of reference gesture q_i

Output:

Predicted gesture label $\hat{y}$

1. Initialize Key Joint Set

Define the set of key joints:

$$K = \{1, 2, \dots, k\}$$

where $k$represents the number of key joints used for classification.

2. Compute Canonical Joint Distances

For each joint $i \in K$, compute the distance between the query joint position $p_i$and the reference joint position q_i:

$$d_i = \| p_i - q_i \|^2$$

3. Calculate Dynamic Joint Weights

Compute the weight for each joint using a normalized exponential function:

where $\alpha$is a hyperparameter controlling the weighting strength.

4. Compute Weighted Cosine Similarity

For each gesture class y, calculate the weighted similarity score:

$$\tilde{o}_y = \sum_{i\in K} \omega_{y,i}\ w_{y,i}^T f_{vi,i}$$

5. Gesture Label Prediction

Select the gesture class with the highest similarity score:

$$\hat{y} = \arg\max_{y\in Y} \quad \tilde{o}_y$$

6. Output

Return the predicted gesture label $\hat{y}$.

C. Loss Function

Several loss functions are used to train the proposed network.

1. Joint Heatmap Loss

The joint heatmap is supervised using the Smooth L1 loss:

$$L_{2D} = \sum_{i\in J} smoothL1(\tilde{p}_i^{2D}, p_i^{2D})$$

where p_i^{2D}represents the ground truth 2D joint location.

2. 3D Joint Estimation Loss

To supervise the JGCN module, a 3D joint regression loss is defined as

$$L_{3D} = \sum_{i\in J} smoothL1(\tilde{p}_i^{3D}, p_i^{3D})$$

3. Canonical Joint Loss

For learning view-independent joint representations, the canonical joint loss is defined as

$$L_{can} = \sum_{i\in K} smoothL1(\tilde{p}_i^{c}, p_i^{c})$$

4.Classification Loss

The classification module is trained using the cross-entropy loss:

$$L_{cls} = -\sum_{i\in Y} o_i \log \frac{\exp(\tilde{o}_i)}{\sum_{j\in Y}\exp(\tilde{o}_j)}$$

Total Loss

The overall training objective is defined as

$$L_{total} = \lambda_1 L_{2D} + \lambda_2 L_{3D} + \lambda_3 L_{can} + \lambda_4 L_{cls}$$

where the weighting parameters are set as

$$\lambda_1 = 2.0,\ \lambda_2 = 2.5,\ \lambda_3 = 2.5,\ \lambda_4 = 1.0.$$

IV. RESULT AND DISCUSSION

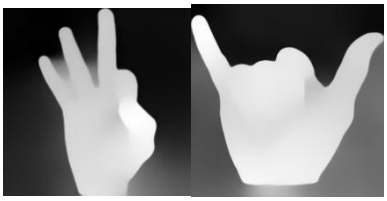


Fig.3 Training Images

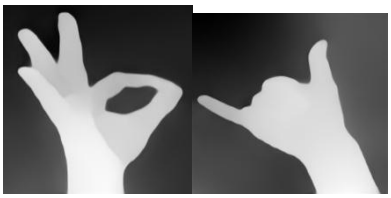


Fig.4 Testing Images

1) Experimental Setup

The proposed framework was implemented using the PyTorch deep learning library. The network parameters were optimized using the AdamW optimizer with a learning rate of 0.0003. During training, the network was first trained until convergence using the base gesture classes. Subsequently, additional gesture classes were introduced in later training sessions to evaluate the adaptability of the model.

Each training epoch consisted of 1000 mini-batches, and the batch size was set to 128. For preprocessing, the hand region was first detected and cropped from the input images following the procedure described in [24]. The cropped images were then resized to a

fixed resolution of 224 × 224 pixels before being used as input to the network. Furthermore, the depth values of the input images were normalized to the range (−1, 1) in order to improve training stability.

For the purpose of joint estimation, the network learns joint representations through the feature extraction and graph-based modules within the proposed architecture. This allows the model to effectively capture both visual features and structural information of hand joints.

2) Dataset

To evaluate the effectiveness of the proposed hand gesture recognition framework, experiments were conducted on a custom hand gesture dataset collected specifically for this study. The dataset consists of five distinct gesture classes, each representing a unique hand configuration. These gestures were selected to provide sufficient variation in finger positioning and hand pose, allowing the model to learn discriminative gesture representations.

Representative examples of the dataset are illustrated in Fig. 1 and Fig. 2. Fig. 1 shows sample training images, which were used to train the proposed model, while Fig. 2 presents sample testing images, which were used to evaluate the performance and generalization capability of the system. The images demonstrate variations in hand orientation, finger placement, and viewpoint, which reflect realistic conditions encountered in practical gesture recognition applications.

The training process was organized into three stages to gradually introduce new gesture classes and evaluate the model's adaptability:

- Base Training Stage: Initially, the model was trained using two gesture classes, which served as the base gestures for learning fundamental gesture features.
- Incremental Session 1: Two additional gesture classes were introduced, increasing the number of recognizable gestures to four classes.
- Incremental Session 2: One additional gesture class was added, resulting in a total of five gesture classes in the final stage.

This staged training procedure enables the evaluation of the proposed framework's ability to maintain stable recognition performance while accommodating additional gesture classes. By progressively expanding the gesture set, the model is tested for its capability to learn new gesture patterns without significantly degrading the recognition accuracy of previously learned gestures.

3) Evaluation Metrics

To evaluate the effectiveness of the proposed hand gesture recognition framework, classification accuracy is used as the primary evaluation metric. Classification accuracy measures the percentage of correctly predicted gesture labels over the total number of test samples and can be expressed as

$$Acc = \frac{No.of\ Correct\ Predictions}{Total\ No.of\ Test\ Samples} \times 100$$

where a higher accuracy indicates better recognition performance.

The evaluation is performed after each training stage by considering all gesture classes that have been introduced up to that stage. In the initial stage, the model is trained using the base gesture classes, and the accuracy obtained at this stage is denoted as acc_0 .As additional gesture classes are introduced in subsequent training sessions, the model is evaluated again, and the final accuracy obtained after the last session is denoted as acc_T.

To analyze how the recognition performance changes as new gesture classes are added, we also compute the performance dropping rate (PD) defined as

$$PD = acc_T - acc_0$$

where

- acc_0represents the classification accuracy after the base training stage, and
- acc_Trepresents the classification accuracy after the final training session.

This metric provides an indication of how well the model maintains its recognition performance when the number of gesture classes increases. A smaller performance drop indicates that the model is capable of maintaining stable classification performance while adapting to additional gesture categories.

In addition to overall accuracy, the classification performance can also be analyzed using confusion matrices, which provide detailed insights into the prediction results for each gesture class. This helps identify potential misclassifications between

visually similar gestures and evaluate the discriminative capability of the proposed method.

Overall, these evaluation metrics allow a comprehensive assessment of the recognition accuracy, stability, and robustness of the proposed hand gesture recognition framework.

C. Gesture Recognition Performance

No.of Gestures	Accuracy (Acc)	Performance drop rate(PD)
2 (Base)	91.67	0
4	89.53	2.14
5	88.33	3.34

The experimental results demonstrate that the proposed framework achieves strong recognition performance across all training stages. During the base training stage, the model successfully learns discriminative representations for the two initial gesture classes, achieving 91.67% of high classification accuracy.

When two additional gestures were introduced in the first incremental session, the proposed method maintained stable performance while effectively learning the newly introduced gestures. This indicates that the feature extraction and graph-based modules are capable of adapting to new gesture variations.

In the second incremental session, the final gesture class was incorporated into the training process. Despite the increased complexity of the classification task, the model continued to maintain high recognition accuracy across all five gesture classes.

Overall, the results demonstrate that the proposed approach is capable of effectively integrating visual features and joint-based structural information, allowing the system to maintain reliable gesture recognition performance as new gesture categories are introduced.

D. Analysis of Viewpoint Variation

Gesture recognition performance can be affected by variations in camera viewpoint and hand orientation. To evaluate the robustness of the proposed method, we analyzed the recognition accuracy under different viewing conditions.

Experimental results show that the proposed method maintains stable performance even when the hand gestures are captured from different viewpoints. This robustness is achieved through the Viewpoint Influence Elimination (VIE) module, which extracts view-independent joint features using graph convolutional networks.

By leveraging both image features and joint-based representations, the model is able to reduce the impact of viewpoint variations and improve generalization across different hand poses.

V. CONCLUSION

In this work, we investigated the problem of open-set hand gesture recognition under unconstrained viewing conditions. The proposed framework focuses on improving the robustness of gesture recognition systems when gestures are captured from different viewpoints or when variations in hand configurations occur. To address these challenges, a method was introduced to reduce viewpoint-related influences in the extracted features, thereby improving the generalization capability of the recognition model. The proposed approach integrates viewpoint influence elimination and joint-based feature representation to enhance the discriminative ability of the model. By utilizing canonical joint positions, the framework calculates dynamic joint weights that improve the effectiveness of cosine similarity during gesture classification. This mechanism allows the system to emphasize the most informative joints and better distinguish between similar gestures. Experimental results demonstrate that the proposed method achieves reliable gesture recognition performance and maintains robustness under viewpoint variations. The results indicate that combining deep visual features with structural joint information significantly improves recognition accuracy compared with conventional approaches. In this study, the focus was primarily on single-hand gesture recognition. Future work will extend this framework to more complex interaction scenarios, including dual-hand gestures and hand–body gesture recognition, which may further enhance the applicability of gesture-based human–computer interaction systems.

REFERENCES

[1] J. Cheng et al., "Skeleton-based gesture recognition with learnable paths and signature features," IEEE Trans. Multimedia, vol. 26, pp. 3951–3961, 2024.

[2] Y. Zhang, C. Cao, J. Cheng, and H. Lu, "EgoGesture: A new dataset and benchmark for egocentric hand gesture recognition," IEEE Trans. Multimedia, vol. 20, no. 5, pp. 1038–1050, May 2018.

[3] J. Wan et al., "ChaLearn looking at people: IsoGD and conGD largescale RGB-D gesture recognition," IEEE Trans. Cybern., vol. 52, no. 5, pp. 3422–3433, May 2022.

[4] D. Avola, M. Bernardi, L. Cinque, G. L. Foresti, and C. Massaroni, "Exploiting recurrent neural networks and leap motion controller for the recognition of sign language and semaphoric hand gestures," IEEE Trans. Multimedia, vol. 21, no. 1, pp. 234–245, Jan. 2019.

[5] S. Sharma and S. Singh, "Vision-based hand gesture recognition using deep learning for the interpretation of sign language," Expert Syst. with Appl., vol. 182, 2021, Art. no. 115657.

[6] B. Zhou, P.Wang, J.Wan, Y. Liang, and F.Wang, "A unified multimodal deand re-coupling framework for RGB-D motion recognition," IEEE Trans. Pattern Anal. Mach. Intell., vol. 45, no. 10, pp. 11428–11442, Oct. 2023.

[7] M. M. Damaneh, F. Mohanna, and P. Jafari, "Static hand gesture recognition in sign language based on convolutional neural network with feature extraction method using ORB descriptor and Gabor filter," Expert Syst. Appl., vol. 211, 2023, Art. no. 118559.

[8] Y. S. Tan, K. M. Lim, and C. P. Lee, "Hand gesture recognition via enhanced densely connected convolutional neural network," Expert Syst. Appl., vol. 175, 2021, Art. no. 114797.

[9] C. Zhang et al., "Few-shot incremental learning with continually evolved classifiers," inProc. 2021 IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2021, pp. 12455–12464.

[10] D.-W. Zhou et al., "Few-shot class-incremental learning by sampling multi-phase tasks," IEEE Trans. Pattern Anal. Mach. Intell., vol. 45, no. 11, pp. 12816–12831, Nov. 2023.

[11] Y. Cui, Z. Yu, W. Peng, Q. Tian, and L. Liu, "Rethinking few-shot classincremental learning with open-set hypothesis in hyperbolic geometry," IEEE Trans. Multimedia, vol. 26, pp. 5897–5910, 2024.

[12] T. Chowdhury et al., "Few-shot class-incremental learning for 3D point cloud objects," in Proc. Comput. Vis., 2022, pp. 204–220.

[13] Z. Song et al., "Learning with fantasy: Semantic-aware virtual contrastive constraint for few-shot class-incremental learning," in Proc. 2023 IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2023, pp. 24183–24192.

[14] C. Peng et al., "Few-shot class-incremental learning from an open-set perspective," in Proc. Eur. Conf. Comput. Vis., 2022, pp. 382–397.

[15] D.-W. Zhou et al., "Forward compatible few-shot class-incremental learning," in Proc. 2022 IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2022, pp. 9046–9056.

[16] L. Zhao et al., "Few-shot class-incremental learning via class-aware bilateral distillation," in Proc. 2023 IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2023, pp. 11838–11847.

[17] Y. Cui et al., "Uncertainty-guided semi-supervised few-shot classincremental learning with knowledge distillation," IEEE Trans. Multimedia, vol. 25, pp. 6422–6435, 2023.

[18] B. Yang et al., "Dynamic support network for few-shot class incremental learning," IEEE Trans. Pattern Anal. Mach. Intell., vol. 45, no. 3, pp. 2945–2951, Mar. 2023.

[19] C. Xu, L. N. Govindarajan, Y. Zhang, and L. Cheng, "Lie-X: Depth image based articulated object pose estimation, tracking, and action recognition on lie groups," Int. J. Comput. Vis., vol. 123, no. 3, pp. 454–478, 2017.

[20] X. Zhang and F. Zhang, "Differentiable spatial regression: A novel method for 3D hand pose estimation," IEEE Trans. Multimedia, vol. 24, pp. 166–176, 2022.

[21] X. Deng et al., "Recurrent 3D hand pose estimation using cascaded poseguided 3D alignments," IEEE Trans. Pattern Anal. Mach. Intell., vol. 45, no. 1, pp. 932–945, Jan. 2023.

[22] H. Li et al., "Hand gesture recognition enhancement based on spatial fuzzy matching

in leap motion," IEEE Trans. Ind. Inform., vol. 16, no. 3, pp. 1885–1894, Mar. 2020.

[23] M. M. Alam, M. T. Islam, and S. M. Rahman, "Unified learning approach for egocentric hand gesture recognition and fingertip detection," Pattern Recognit., vol. 121, 2022, Art. no. 108200.

[24] F. Xiong et al., "A2J: Anchor-to-joint regression network for 3D articulated pose estimation from a single depth image," in Proc. Int. Conf. Comput. Vis., Oct. 2019, pp. 793–802.

[25] Y.-S. Hsiao, J. Sanchez-Riera, T. Lim, K.-L. Hua, and W.-H. Cheng, "Lared: A large RGB-D extensible hand gesture dataset," in Proc. 5th ACM Multimedia Syst. Conf., New York, NY, USA, 2014, pp. 53–58.

[26] C. Xu et al., "Robust 3D hand detection from a single RGB-D image in unconstrained environments," Sensors, vol. 20, no. 21, 2020, Art. no. 026520.

[27] N. Pugeault and R. Bowden, "Spelling it out: Real-time asl fingerspelling recognition," in Proc. 2011 IEEE Int. Conf. Comput. Vis. Workshops, 2011, pp. 1114–1119.

[28] L. Minto, G. Marin, and P. Zanuttigh, "3D hand shape analysis for palm and fingers identification," in Proc. 11th IEEE Int. Conf. Workshops Autom. Face Gesture Recognit., 2015, pp. 1–6.

[29] P. P. Kumar, P. Vadakkepat, and A. P. Loh, "Hand posture and face recognition using a fuzzy-rough approach," Int. J. Humanoid Robot., vol. 07, no. 03, pp. 331–356, 2010.

[30] H. Cheng, L. Yang, and Z. Liu, "Survey on 3D hand gesture recognition," IEEE Trans. Circuits Syst. Video Technol., vol. 26, no. 9, pp. 1659–1673, Sep. 2016.

[31] Z. Yu et al., "Searching multi-rate and multi-modal temporal enhanced networks for gesture recognition," IEEE Trans. Image Process., vol. 30, pp. 5626–5640, 2021.

[32] C. Wang, Z. Liu, and S.-C. Chan, "Superpixel-based hand gesture recognition with kinect depth camera," IEEE Trans. Multimedia, vol. 17, no. 1, pp. 29–39, Jan. 2015.

[33] N. H. Dardas and N. D. Georganas, "Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques," IEEE Trans. Instrum. Meas., vol. 60, no. 11, pp. 3592–3607, Nov. 2011.

[34] S. P. Priyal and P. K. Bora, "A robust static hand gesture recognition system using geometry based normalizations and Krawtchouk moments," Pattern Recognit., vol. 46, no. 8, pp. 2202–2219, 2013.

[35] P. K. Pisharady, P. Vadakkepat, and A. P. Loh, "Attention based detection and recognition of hand postures against complex backgrounds," Int. J. Comput. Vis., vol. 101, no. 3, pp. 403–419, Feb. 2013.

[36] Y. Li, X. Wang, W. Liu, and B. Feng, "Deep attention network for joint hand gesture localization and recognition using static RGB-D images," Inf. Sci., vol. 441, pp. 66–78, 2018.

[37] A. Bandini and J. Zariffa, "Analysis of the hands in egocentric vision: A survey," IEEE Trans. Pattern Anal. Mach. Intell., vol. 45, no. 6, pp. 6846–6866, Jun. 2023.

[38] D. Avola et al., "3D hand pose and shape estimation from RGB images for keypoint-based hand gesture recognition," Pattern Recognit., vol. 129, 2022, Art. no. 108762.

[39] S. Sharma and S. Huang, "An end-to-end framework for unconstrained monocular 3D hand pose estimation," Pattern Recognit., vol. 115, 2021, Art. no. 107892.

[40] C. Xiao, N. Madapana, and J. Wachs, "One-shot image recognition using prototypical encoders with reduced hubness," in Proc. 2021 IEEE/CVF Winter Conf. Appl. Comput. Vis., Jan. 2021, pp. 2252–2261.