A Novel Genetic Programming Approach for Skeleton-Based Human Action Recognition

Youkai Xiao[1], Mitchell Rogers[1], Patrice Delmas[2], Bing Xue[1], Mengjie Zhang[1]
[1]*Centre for Data Science and Artificial Intelligence & School of Engineering and Computer Science*
Victoria University of Wellington, Wellington, New Zealand
{youkai.xiao, bing.xue, mengjie.zhang}@ecs.vuw.ac.nz, mitchell.rogers@vuw.ac.nz
[2]*IVSLab, The University of Auckland, Auckland, New Zealand*
p.delmas@auckland.ac.nz

***Abstract*—Human action recognition (HAR) plays a vital role in many real-world applications, such as sports. However, illumination changes and complex backgrounds make video-based HAR highly challenging. Skeleton-based HAR (SHAR) alleviates these issues by representing human motion using structured joint coordinates, which are less sensitive to variations in illumination and background. Despite recent progress, popular deep neural network based approaches to SHAR often rely on large-scale labeled data and offer limited interpretability. To address these limitations, we explore genetic programming (GP) as an alternative that can automatically construct discriminative, human-interpretable motion descriptors from skeleton sequences. In this work, we propose a GP-based approach, named SkeletonGP, which extracts informative geometric, motion, statistical, and correlation features via a multi-layer feature extraction framework. The performance of our approach is evaluated on a popular skeleton-based HAR dataset (Penn Action dataset). The results suggest that the proposed SkeletonGP approach achieves competitive performance compared to deep neural network methods, while providing a promising interpretable solution. Furthermore, the ablation study demonstrates the importance of adding a statistical and correlation feature extraction layer on top of the geometric and motion feature extraction layer.**

***Index Terms*—Skeleton-based Human Action Recognition, Genetic Programming (GP), Multi-layer Feature Extraction.**

I. Introduction

HUMAN action recognition (HAR) is the task of recognizing human actions from a set of predefined categories based on sensor observations (e.g., videos, depth maps, skeleton sequences), by modeling the spatio-temporal dynamics of human actions [1]. HAR is a highly active research area in computer vision and has a wide range of real-world applications, such as sports [2], human-computer interaction [3], video surveillance [4], robotics [5], and healthcare [6]. This topic has been investigated using different types of data, including RGB videos, depth videos, and body skeleton joints. Skeleton-based human action recognition (SHAR) has gained significant attention because it is not affected by background, clothing, or lighting, and it provides precise body pose information in the form of 2D or 3D coordinates of the body joints. In recent years, rapid advances in AI techniques have made it much easier to extract accurate human-skeleton data directly from video without relying on additional sensors [1].

Skeletons can be considered as graphs, where the vertices correspond to the skeleton joints and the edges represent the connections between joints. Since Graph Convolutional Networks (GCNs) were introduced [7], Chen et al. [8] demonstrated the effectiveness of GCNs in capturing spatio-temporal skeleton features. The remarkable success of the Transformer [9] for modeling sequential data through the self-attention mechanism has inspired numerous studies to adopt transformers for SHAR. Transformer-based methods assign attention weights to different joints and frames, which allow the model to focus on the most representative spatial and temporal parts in each sequence[10], [11]. In certain cases, attention-based methods are combined with GCNs to achieve robust action representation [11], [12].

Although deep neural network–based (DNN-based) methods have achieved strong performance in SHAR, they suffer from two key limitations. Firstly, they typically require a large amount of labelled training data. However, large-scale annotation is often costly, particularly in sports scenarios and other domain-specific scenarios, where accurately labelling professional actions demands domain-specific expertise. Secondly, owing to their black-box nature, these methods provide limited interpretability. In sports action recognition, interpretability is often important for coaching staff, as it enables targeted analysis and refinement of athletes' movements.

Genetic programming (GP) offers a promising solution. Compared with DNN-based approaches, the potential interpretability of GP makes it well suited for analysing actions in interpretability-critical scenarios. Moreover, GP's flexibility and feature-extraction capability suggest strong potential for extracting and constructing effective information from skeleton data. Therefore, this paper proposes a novel GP approach to SHAR. The main contributions of this paper are summarized as follows:

- To the best of our knowledge, this is the first attempt to apply GP to skeleton-based human action recognition, which can achieve competitive recognition accuracy while also exhibiting promising interpretability.
- A novel GP approach is proposed, which adopts a new representation, along with a new function set and a new terminal set, to extract multi-level features from skeleton data, including geometric and motion features such as angles, velocities, accelerations, and displacements, as well as statistical and correlation features derived from these geometric and motion features.

- To examine the effectiveness of the proposed method, we compare it against mainstream DNN-based SHAR methods on a popular SHAR dataset.

II. RELATED WORK

A. Traditional Machine Learning and DNN-based Methods for SHAR

Early studies employed machine learning to extract handcrafted features from skeleton joint sequences, typically including joint motion trajectories, spatio-temporal descriptors, and rotation angles. For motion trajectories, Hussein et al. [13] partitioned each skeleton sequence into multiple subsequences and computed covariance matrices to capture trajectory-related joint dynamics. For spatio-temporal features, Xia et al. [14] proposed using histogram-based representations to model the spatio-temporal position variations of the arms and legs, thereby mining fine-grained motion details from the underlying features. For rotation-angle features, Alan et al. [15] computed key-joint angles from Kinect skeleton data and combined them with the relative positions among the head, spine, and hands to characterize limb motion trends, then organized continuous action representations into action sequences for classification. Although these machine learning methods achieve good performance in action recognition, their reliance on handcrafted features imposes clear limitations in terms of generalization and efficiency.

DNN-based methods currently dominate the field of SHAR, with mainstream approaches primarily built on GCNs and attention-based architectures. Yan et al. [16] introduced ST-GCN to model the spatial and temporal relationships among the joints in a skeleton sequence. This pioneering work marked the first successful application of GCNs for skeleton-based action recognition, and became the main component for numerous methods.

A Transformer-based action recognition method can globally model the dynamic variations of joints within skeleton sequences, enabling more precise representation of joint interactions throughout the action [17]. Plizzari et al. [18] proposed a spatio-temporal Transformer network that decouples spatial and temporal dimensions of human motion. The spatial attention module learns the intra-frame joint interactions, while the temporal attention module models the inter-frame motion dynamics.

Although DNN-based methods have achieved strong performance in SHAR, they still suffer from a reliance on large-scale labeled data and limited interpretability.

B. GP for Human Action Recognition

Liu et al. [19] proposed a GP-based method to automatically learn spatio-temporal motion descriptors for action recognition, rather than handcrafted features. GP evolves multi-layer feature descriptors from a predefined set of primitive 3D operators (e.g., 3D-Gabor and wavelet) to extract scale- and shift-invariant representations from both color and optical-flow sequences. This work demonstrates the potential of GP for action recognition. However, directly processing color and optical-flow sequences leads to a high computational cost and a much longer GP runtime. In addition, there has not been any GP approach with skeleton features reported to date.

III. PROPOSED METHOD

In this section, we first present the overall algorithm. We then detail the architecture of the new GP-based framework, followed by a description of the proposed function set, terminal set, and the fitness function used for evaluation. Finally, we report the experimental setup and results.

A. Algorithm Overview

In SkeletonGP, firstly, an initial population of GP individuals are randomly generated using ramped half-and-half for program generation based on the SkeletonGP representation. Each individual extracts a set of high-level features from the input skeletons, and then the features are fed into a linear support vector machine (SVM) to predict the action class. The linear SVM will perform a 5-fold cross-validation on the training set, and the average value of the classification accuracy is used as the fitness value of that individual program. After completing the fitness evaluation, parent individuals will be selected by tournament selection and then the offspring will be generated by genetic operators such as subtree crossover, subtree mutation, and elitism. This process is repeated for a number of generations, and finally the best individual program is returned as the trained model.

In the testing process, the high-level features are first obtained by feeding the test set samples to the GP solution. Then the extracted high-level features are fed into the linear SVM to obtain the classification results using the same process as training.

B. Program Structure

Fig. 1 illustrates the proposed tree structure and an example GP individual, where different colours indicate different layers. The SkeletonGP tree structure is built based on strongly typed GP (STGP)[20], which is capable of evolving programs with functions or terminals that operate on the appropriate data types. Based on STGP, SkeletonGP adopts a multi-layer tree structure comprising an input layer, a geometric and motion feature extraction layer, an optional statistical and correlation feature extraction layer, a feature concatenation layer, and an output layer, with different function types assigned to each layer. The terminals are joint coordinate sequence (JCS) extracted from a skeleton sample, and each leaf node is a single joint's $T \times 2$ coordinate sequence, where T is the number of frames. The geometric and motion layer derives joint-to-joint distances and their variations, as well as joint angles and frame-to-frame angle changes, from the input joint coordinate sequence (JCS). The statistical layer, highlighted by the dashed box in Fig. 1, further computes segment-wise max/min/mean statistics and inter-feature dependency measures, including Pearson correlation, covariance, and mutual information, on the geometric/motion outputs. This layer may exist or be absent in SkeletonGP trees, allowing SkeletonGP to output only geometric/motion features for simpler tasks or to combine

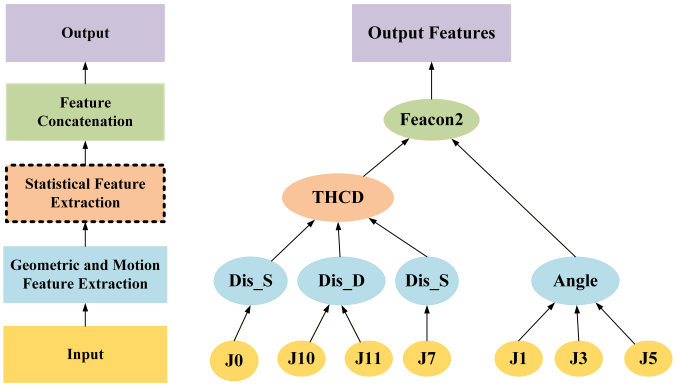

Fig. 1. The new program structure of SkeletonGP and an example program of SkeletonGP.

them with statistical features for more complex recognition. Finally, the feature concatenation layer merges the features from child nodes into a single feature vector, which will be fed into a linear SVM for classification. =

C. Terminal Set

In SkeletonGP, the terminal set contains a single type, namely the JCS extracted from each skeleton sample. A skeleton sample is a temporal sequence of T frames, each providing 2D coordinates for a set of body joints. For each joint, we represent its trajectory as a $T \times 2$ JCS, and each leaf node corresponds to one joint's $T \times 2$ sequence. This design offers two key benefits: 1) it supports subsequent geometric, motion, statistical, and correlation feature extraction; and 2) it improves interpretability by allowing explicitly revealing which joint-based distances, angles, and related features contribute to recognition.

D. Function Set

There are three different types of functions in the function set of SkeletonGP: geometric and motion feature extraction functions, statistical and correlation feature extraction functions, and feature concatenation functions. The input and output data types of the three different types of functions, along with their brief descriptions, are summarized in Tables I, II, and III. The followings are detailed introductions to these operators.

Distance-Different: This function randomly selects two joints from all joints in a skeleton sequence and computes the Euclidean distance between them at each frame. The dimensionality of the output feature vector equals the number of frames in the sample. Frame-wise distances between 2 different joints can capture the temporal dynamics of human action.

Distance-Same: For a given joint, this function computes Euclidean distances between consecutive frames $(t,\ t+1)$ and between the first and last frames, producing an output feature vector with length equal to the number of frames. These distances directly quantify the joint's motion magnitude.

TABLE I
GEOMETRIC AND MOTION FEATURE EXTRACTION FUNCTIONS

Functions	Input	Output	Description
Distance-Different	2 different JCSs	Vector	Frame-wise distance between 2 different joints
Distance-Same	1 JCS	Vector	Distance of the same joint between different frames
Angle	3 different JCSs	Vector	Frame-wise angle between 3 different joints
Angle-delta	3 different JCSs	Vector	Frame-to-frame angle difference among three joints
Distance-delta	1 JCS	Vector	Second-order frame-to-frame distance of the same joint
Distance-diff	2 different JCSs	Vector	Frame-wise distance variation between 2 joints

TABLE II
STATISTICAL AND CORRELATION FEATURE EXTRACTION FUNCTIONS

Functions	Input	Output	Description
Segment-wise-Max	Vector	Vector	The maximum values of FSEC
Segment-wise-Min	Vector	Vector	The minimum values of FSEC
Segment-wise-Mean	Vector	Vector	The mean values of FSEC
Covariance	2 Vectors	Vector	Calculate the covariance of FSEC of the input two vectors.
Pearson-correlation	2 Vectors	Vector	Calculate the Pearson correlation of FSEC of the input two vectors
Mutual-Information	2 Vectors	Vector	Calculate the mutual information of FSEC of the input two vectors
THCD	3 Vectors	Vector	Calculate the segment-wise covariance matrix among the three vectors

Note: FSEC is an abbreviation for "the first half, the second half, and the entire sequence".

Angle: This function calculates the angle formed by three different joints for every frame in the sequence. The dimensionality of the output feature vector equals the number of frames in the sample.

Angle-delta: This function computes the per-frame angle defined by three joints and then takes frame-to-frame angle differences $(t,\ t+1)$, plus a first-last difference, producing a vector whose length equals the number of frames. These differences capture joint flexion-extension speed and help distinguish actions with similar poses but different angular change rates (e.g., slow vs. fast running).

Distance-delta: This function computes frame-to-frame Euclidean distances for a given joint and then takes differences between consecutive distances, and a first-last difference, producing a feature vector whose length equals the number of frames. The resulting second-order distance acts as an acceleration-like cue, capturing higher-order joint dynamics that are informative for explosive or rapid movements.

Distance-diff: This function computes the per-frame Euclidean distance between two joints and then takes frame-to-frame differences, and a first-last difference, producing a vector whose length equals the number of frames and capturing distance variation over the action.

Segment-wise-Max, ***Segment-wise-Min***, and ***Segment-wise-Mean*** compute the maximum, minimum, and mean val-

ues over the first half, second half, and the entire input vector, respectively. These statistics summarize phase-wise and global extrema and averages of angle/displacement sequences, capturing peak and low-amplitude configurations (e.g., most flexed/contracted states) and average motion intensity in different temporal stages, producing noise-robust and discriminative features for skeleton-based action recognition.

Covariance: This function computes the covariance between two input vectors over the first half, the second half, and the full sequence. The resulting segment-wise covariances capture inter-joint motion coordination (e.g., between angles and their velocities), which is highly discriminative for action recognition. For instance, kicking often exhibits strong covariance between leg angle and leg angular velocity.

Pearson-correlation: This function computes Pearson correlation coefficients between two input vectors over the first half, the second half, and the full sequence. These segment-wise correlations quantify whether the two feature sequences vary synchronously (positive or negative), providing robust and discriminative cues for skeleton-based action recognition.

Mutual-Information: This function computes mutual information [21] between two input vectors over the first half, the second half, and the full sequence. Unlike covariance and Pearson correlation, which capture only linear dependence, mutual information measures nonlinear inter-joint coordination and is therefore well suited to complex actions.

Temporal Hierarchical Covariance Descriptor: This function computes covariance matrices over three temporal segments (first half, second half, and full sequence) for three feature vectors (e.g., angles or displacements). For each symmetric matrix, only the six upper-triangular elements are retained, yielding an 18-dimensional output across the three segments.

FeaCon2 and ***FeaCon3*** are able to connect two and three feature vectors from the feature extraction layer into one feature vector, respectively. Additionally, *FeaCon2* can be used as an input to either *FeaCon2* or *FeaCon3*, and the same goes for *FeaCon3*.

TABLE III
FEATURE CONCATENATION FUNCTIONS

Functions	Input	Output	Description
Feacon2	2 Vectors	Vector	Concatenate two vectors into one feature vector
Feacon3	3 Vectors	Vector	Concatenate three vectors into one feature vector

E. Fitness Evaluation

SkeletonGP uses K-fold cross-validation on the training set to evaluate each GP individual. The training set is divided into K folds, where $K-1$ folds (referred to as the evaluation training set) are used to train a classifier, while the remaining fold (the evaluation test set) is used to evaluate the trained classifier. This process is repeated K times. The fitness of each GP individual is calculated as the average accuracy across all K evaluation test sets. Due to the limited size of the training set, K is set to 5.

IV. Experiment Setup

To verify the proposed method, a series of experiments are designed in this section. This section introduces the dataset, the setup of the comparison methods and the related parameter settings.

A. Datasets

Experiments were performed on the Penn Action dataset [22]. The Penn Action dataset contains 15 action classes (e.g., baseball swing, golf swing, and tennis serve) and 2,326 RGB video sequences collected from multiple sports, each with manually annotated 2D skeleton joint keypoints, and is widely used for SHAR.

Since the original Penn Action dataset contains sequences of varying lengths, we perform a preprocessing step to facilitate model training and evaluation. All original samples contain more than 16 frames, so we uniformly sample 16 frames from each sequence to construct a new dataset with sequences of same length. The official split of the Penn Action dataset consists of 1,258 training samples and 1,068 test samples.

B. Methods for Comparisons

We compare our approach against various mainstream skeleton-based action recognition approaches on the Penn Action dataset, including ST-GCN [16], 2S-AGCN [23], MS-G3D [24], and UNIK [25]. ST-GCN is the first work to apply GCNs to SHAR and has become a widely used baseline. 2s-AGCN introduces adaptive graph learning (learnable adjacency) and a two-stream joint/bone architecture, where the bone stream uses adjacent-joint difference vectors and the two streams are fused for improved accuracy. MS-G3D proposes a unified spatiotemporal graph convolution (G3D) operator and combines it with multi-scale graph convolutions to capture long-range dependencies more effectively. UNIK learns data-driven spatiotemporal relations via learnable dependency matrices within a unified framework, aiming to improve generalization beyond fixed, hand-crafted graph topologies.

C. Parameter Settings

In SkeletonGP, the size of the population is 200 and the maximal number of generations is 50. The depth of the individuals in the initialised population is from 2 to 6. SkeletonGP initializes the population using the ramped half-and-half population initialization strategy. During the evolutionary process, bloat may occur, that is, the size of the individuals keeps increasing but the performance is not improved. Therefore, the maximal depth of individuals in the evolution process is set to 8 to alleviate this problem. Tournament selection with a tournament size of 5. For the genetic operators, the crossover probability is 0.8, the mutation probability is 0.19, and the elitism probability is 0.01. These parameter values are set based on common practice [26]. The proposed algorithm is independently run 30 times. Unless otherwise stated, all settings for the compared methods follow their original papers [24], [23], [16], [25].

V. Results and Discussions

A. Ablation Study

We evaluate the effect of the statistical layer on the Penn Action dataset by comparing the proposed method with the statistical layer against a variant without using the statistical layer. All experiments are independently repeated 30 times. A Wilcoxon rank-sum test is conducted at the 5% significance level to evaluate the statistical significance of the performance improvements. For the Wilcoxon test results, "+" indicates that proposed SkeletonGP performs significantly better than the variant without using the statistical layer.

TABLE IV
Ablation experiment results on the Penn Action dataset.

Methods	Accuracy(%)(Mean ± Std)
SkeletonGP(without statistical feature layer)	70.53±5.87+
SkeletonGP	**84.28±2.67**

From the results in Table IV, the proposed method achieves nearly 14% higher mean accuracy than the variant without the statistical layer over 30 runs on the Penn Action dataset. This result is reasonable, since an action typically involves coordinated movements across multiple joints; therefore, extracting inter-joint coordination features on top of the first-layer geometric and motion features is necessary.

B. Comparison with DNN-based Methods

In the Table V, "*" denotes SkeletonGP's best accuracy over 30 independent runs, while "†" denotes the mean accuracy and standard deviation. It is important to note that we used the best result of SekletonGP here, since those DNN-based methods mostly reported the best results (or those papers did not explicitly state whether the results are the best among multiple runs or that of a single run). As shown in Table V, SkeletonGP achieves a best accuracy of 88.05% using JCS as input, and its performance is highly competitive to deep neural network methods that also rely solely on joint data. This result demonstrates that the proposed GP-based framework can effectively capture discriminative motion patterns.

TABLE V
Comparison with baseline methods on the Penn Action dataset.

Methods	Modality	Accuracy (%)
ST-GCN [16]	Joint	89.6
2s-AGCN [23]	Joint	89.5
MS-G3D [24]	Joint	88.5
UNIK [25]	Joint	90.1
SkeletonGP (Ours)	**Joint**	**88.05***
SkeletonGP (Ours)	**Joint**	**84.28±2.67**†

C. Further Analysis

This GP individual program constructs a spatio-temporal motion descriptor from selected joints (J0–J12). In the Penn Action dataset, the mapping between joint indices and joint names is summarized in Table VI. As shown by the evolved GP program in Fig. 2, in the geometric and motion feature extraction layer, the best individual employs inter-frame displacements of the left/right wrists (J5/J6) and their temporal differences, inter-frame displacements of the left/right ankles (J11/J12), and shoulder–elbow–wrist angles for both arms. These primitives are effective because wrist-end trajectories are prominent in actions in baseball swing/pitch, tennis forehand/serve, and guitar strumming, and fine-grained differences are often reflected in wrist motion amplitude and peak speed.

TABLE VI
Joint index mapping in Penn Action

Idx	Joint	Idx	Joint	Idx	Joint
J0	Head	J5	Left wrist	J9	Left knee
J1	Left shoulder	J6	Right wrist	J10	Right knee
J2	Right shoulder	J7	Left hip	J11	Left ankle
J3	Left elbow	J8	Right hip	J12	Right ankle
J4	Right elbow				

In the statistical and correlation feature extraction layer, the best individual employs the Pearson correlation between the inter-frame displacement sequences of the left ankle (J11) and right ankle (J12), together with the segment-wise maximum of the temporal difference of the right-wrist (J6) inter-frame distance, which approximates stage-wise peaks in wrist acceleration. These operators are effective because many Penn Action classes are symmetric (e.g., jumping jacks, jump rope, bench press, and squat), where left–right ankle coordination is discriminative. They are also informative for swing- and serve-type actions (e.g., baseball swing/pitch, golf swing, and tennis serve), where stage-wise right-wrist peak acceleration captures fine-grained motion dynamics.

By evolving compact symbolic programs instead of optimizing large-scale neural networks (the mainstream model UNIK contains approximately 3.45 million parameters), SkeletonGP significantly reduces model complexity and computational requirements while achieving competitive recognition performance. Unlike DNN-based models that rely on gradient-based optimization and GPU-intensive training, SkeletonGP employs an evolutionary search process, reducing reliance on large-scale labeled data and expensive computational resources. This capability is particularly promising for deployment in real-world, resource-limited environments, such as edge devices deployed in sports venues or wearable systems, where computational resources and labeled training data are both constrained.

VI. Conclusions

The goal of this paper is to develop a novel GP-based method for SHAR. This goal has been successfully achieved by developing a new GP program structure of multiple feature extraction layers, including a geometric and motion feature extraction layer, an optional statistical and correlation feature extraction layer, and designing a new function set with various operators (e.g., angle and distance computations), and using joint coordinate sequences as the terminal set. The new method was examined and compared with several modern DNN-based methods on the Penn Action action dataset, which is regarded as a benchmark for SHAR. The results suggest that the proposed GP approach, SkeletonGP, achieved very

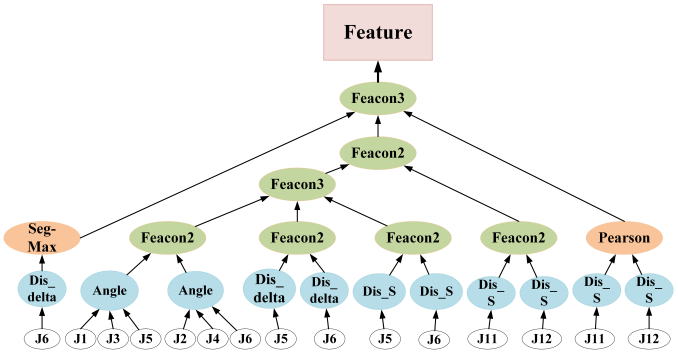

Fig. 2. An example program of SkeletonGP on the experiment of Penn Action dataset.

competitive accuracy performance to the recent DNN-based methods. More importantly, the evolved recognition models by the new GP method are much more interpretable than those learned by deep neural networks with millions of parameters.

A key limitation of DNN-based methods is their black-box nature, which makes it difficult to interpret the decision process despite strong predictive performance. In contrast, SkeletonGP outputs an explicit symbolic program built from interpretable primitives (e.g., joint angles, distances, and velocity-related measures), enabling fine-grained analysis of joint-level motion patterns. This interpretability can help coaches and athletes identify the most influential inter-joint angles, joint-wise motion magnitudes and accelerations, and coordination patterns within an action, providing valuable cues for performance analysis.

In the literature, we have found that dual-stream deep neural networks have used both the joint and the bone modalities for improving the performance. As a future work, we will extend SkeletonGP to a multi-tree GP framework. In this framework, separate trees will extract joint- and bone-based features, and an additional fusion tree will explicitly combine features across modalities. This extension is expected to further improve recognition accuracy while preserving the interpretability and efficiency of the proposed approach.

Acknowledgement

This work was supported in part by the MBIE Endeavor Smart Ideas Project under contracts UOAX2302.

References

[1] J. Xu, Y. Guo, and Y. Peng, "Finepose: Fine-grained prompt-driven 3d human pose estimation via diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 561–570.

[2] Z. Zhao, W. Chai, S. Hao, W. Hu, G. Wang, S. Cao, M. Song, J.-N. Hwang, and G. Wang, "A survey of deep learning in sports applications: Perception, comprehension, and decision," *IEEE Transactions on Visualization and Computer Graphics*, vol. 31, no. 10, pp. 9368–9386, 2025.

[3] X. Chen, X. Luo, J. Weng, W. Luo, H. Li, and Q. Tian, "Multi-view gait image generation for cross-view gait recognition," *IEEE Transactions on Image Processing*, vol. 30, pp. 3041–3055, 2021.

[4] M. Ding, Y. Ding, L. Wei, Y. Xu, and Y. Cao, "Individual surveillance around parked aircraft at nighttime: Thermal infrared vision-based human action recognition," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 2, pp. 1084–1094, 2022.

[5] C. Papaioannidis, I. Mademlis, and I. Pitas, "Fast cnn-based single-person 2d human pose estimation for autonomous systems," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 3, pp. 1262–1275, 2022.

[6] Y. Wei, X. Chen, and J. Gu, "Human activity recognition soc for ar/vr with integrated neural sensing, ai classifier and chained infrared communication for multi-chip collaboration," in *2023 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits)*. IEEE, 2023, pp. 1–2.

[7] T. Kipf, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[8] Y. Chen, Z. Zhang, C. Yuan, B. Li, Y. Deng, and W. Hu, "Channel-wise topology refinement graph convolution for skeleton-based action recognition," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 13 359–13 368.

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[10] A. F. Bavil, H. Damirchi, and H. D. Taghirad, "Action capsules: Human skeleton action recognition," *Computer Vision and Image Understanding*, vol. 233, p. 103722, 2023.

[11] Y. Pang, Q. Ke, H. Rahmani, J. Bailey, and J. Liu, "Igformer: Interaction graph transformer for skeleton-based human interaction recognition," in *European Conference on Computer Vision*. Springer, 2022, pp. 605–622.

[12] C. Si, W. Chen, W. Wang, L. Wang, and T. Tan, "An attention enhanced graph convolutional lstm network for skeleton-based action recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 1227–1236.

[13] M. E. Hussein, M. Torki, M. A. Gowayyed, and M. El-Saban, "Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations." in *Ijcai*, vol. 13, 2013, pp. 2466–2472.

[14] L. Xia, C.-C. Chen, and J. K. Aggarwal, "View invariant human action recognition using histograms of 3d joints," in *2012 IEEE computer society conference on computer vision and pattern recognition workshops*. IEEE, 2012, pp. 20–27.

[15] C. Wang, Y. Wang, and A. L. Yuille, "An approach to pose-based action recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 915–922.

[16] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[17] H. Liu, Y. Liu, Y. Chen, C. Yuan, B. Li, and W. Hu, "Transkeleton: Hierarchical spatial–temporal transformer for skeleton-based action recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 8, pp. 4137–4148, 2023.

[18] C. Plizzari, M. Cannici, and M. Matteucci, "Spatial temporal transformer network for skeleton-based action recognition," in *International conference on pattern recognition*. Springer, 2021, pp. 694–701.

[19] L. Liu, L. Shao, X. Li, and K. Lu, "Learning spatio-temporal representations for action recognition: A genetic programming approach," *IEEE transactions on cybernetics*, vol. 46, no. 1, pp. 158–170, 2015.

[20] D. J. Montana, "Strongly typed genetic programming," *Evolutionary computation*, vol. 3, no. 2, pp. 199–230, 1995.

[21] T. M. Cover, *Elements of information theory*. John Wiley & Sons, 1999.

[22] W. Zhang, M. Zhu, and K. G. Derpanis, "From actemes to action: A strongly-supervised representation for detailed action understanding," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 2248–2255.

[23] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 12 026–12 035.

[24] Z. Liu, H. Zhang, Z. Chen, Z. Wang, and W. Ouyang, "Disentangling and unifying graph convolutions for skeleton-based action recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 143–152.

[25] D. Yang, Y. Wang, A. Dantcheva, L. Garattoni, G. Francesca, and F. Brémond, "Unik: A unified framework for real-world skeleton-based action recognition," *arXiv preprint arXiv:2107.08580*, 2021.

[26] Y. Bi, B. Xue, and M. Zhang, "Genetic programming with image-related operators and a flexible program structure for feature learning in image classification," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 1, pp. 87–101, 2020.