View-Adaptive Graph Neural Network for Action Recognition

This paper was downloaded from TechRxiv (<https://www.techrxiv.org>).

LICENSE

CC BY-NC-SA 4.0

SUBMISSION DATE / POSTED DATE

13-01-2022 / 19-01-2022

CITATION

Shahid, Ali Raza; Nawaz, Mehmood; YAN, Hong; Fan, Xinqi (2022): View-Adaptive Graph Neural Network for Action Recognition. TechRxiv. Preprint. <https://doi.org/10.36227/techrxiv.18318929.v1>

DOI

[10.36227/techrxiv.18318929.v1](https://doi.org/10.36227/techrxiv.18318929.v1)

View-Adaptive Graph Neural Network for Action Recognition

Ali Raza Shahid, *Student Member, IEEE*, Mehmood Nawaz, *Member, IEEE*, Xinqi Fan, *Student Member, IEEE*, Hong Yan, *Fellow, IEEE*

Abstract—Skeleton-based recognition of human actions has received attention recently because of popular 3D acquisition sensors. Existing studies use skeleton data from video clips collected from several views. When humans perform certain actions, the body view shifts from the camera perspective, resulting in unstable and noisy skeletal data, and the possibility of self-occlusions makes it very challenging for recognition. To counteract the influence of variations, we developed a view-adaptive mechanism that identifies the viewpoints across the action video sequence and transforms the skeleton view through a data-driven learning process. Most existing methods use a fixed human-defined prior criterion to re-position the skeletons. In contrast, we utilize an unsupervised re-position approach and jointly design a view-adaptive (VA) neural network based on the graph neural network (GNN). Our VA-GNN model can transform the skeletons of distinct views into more consistent virtual perspective than comparative pre-processing approaches. The VA module learns the best-observed view because it determines the most suitable view and transforms the skeletons from action sequence for end-to-end recognition along with suited graph topology with adaptive GNN. Experiments on three datasets achieve a better accuracy and a lower number of parameters demonstrating the effectiveness of our approach.

Index Terms—Action recognition, 3D skeleton, View-adaptive, graph convolution neural network.

I. INTRODUCTION

Human action recognition using visual data is one of the important research areas in computer vision in recent years. The human-computer applications are related to video understanding, games interaction, and surveillance systems [1]. There are different kinds of action data collection modalities, but most methods are based on RGB video data. Action recognition can be recognized efficiently using the 3D skeleton joints on the human body, which has recently gained popularity [2]. The representation of video data as a skeleton is more robust in variations during acquisition viewpoints and

distractions in the surrounding area of the human body [3], [4], [5]. The motion of a few joints on the body without the appearance information identifies human actions [6]. With the advent of depth acquisition sensors such as two or more RGB camera devices, StereoLabs Zed2 [7], RealSense [8], Kinect [9], and advance methods to detect humans with an estimation of the pose [10], made a collection of 3D skeleton data easy. This paper focuses on 3D skeleton-based action recognition.

Fig. 1. RGB and 3D skeleton representation of (a) thumbs up action at different viewpoints and (b) same human body posture captured from three different viewpoints simultaneously.

One of the major challenges in real-world applications is different viewpoints for human action recognition while observing the sequenced video. The examples of various viewpoints are illustrated in Fig. 1. The actor performing the action changes the pose, which impacts the recognition in terms of variation [11]. For the same human pose, the posture of a skeleton can be changed while capturing the data. From an observation point of view, this variation makes action recognition very challenging [12]. The varied samples may not be included in the test sample, which can cause a significant decrease in the recognition performance. To handle such a problem of variation in viewpoint, a large number of datasets and models with huge parameters are usually considered. However, the large model is very difficult to train, and it's prone to overfitting. For robust action recognition under the view variation problem, many attempts have been made in the previous works as in [13], [14], [15], [16], [17], [18], [12], but most of the previous works are designed for RGB video data. Therefore, they require a large amount of data from different views or design an invariant feature description that does not provide satisfactory performance. Motivated by the fact that a study of view

Manuscript submitted Jan 22, 2021. This work is supported by the Hong Kong Innovation and Technology Commission and Hong Kong Research Grants Council (Project CityU 11204821). (Corresponding author: Ali Raza Shahid)

A. R. Shahid is with Department of Electrical Engineering, City University of Hong Kong, Kowloon 100071, Hong Kong (e-mail: a.raza@my.cityu.edu.hk, aliraza_ee@yahoo.com)

H. Yan is with Department of Electrical Engineering, City University of Hong Kong, Kowloon 100071, Hong Kong (e-mail: h.yan@cityu.edu.hk).

M. Nawaz is Post Doc-Fellow at Chinese University of Hong Kong

X. Fan is with Department of Electrical Engineering, City University of Hong Kong

Color figures of one or more of the figures in this paper are available with wider content information online at <http://ieeexplore.ieee.org>. (paper

adaptive or invariance feature description in skeleton-based human action recognition is still at early stages.

Previous skeleton-based action recognition works have made few attempts to account for the effect of view variations. To make the human skeleton invariant to absolute location and body orientation, a pre-processing is usually used as in [2], [19], [20], [21], [22], [23], [24], [25], [26]. These methods employ 3D coordinate transformation to represent a person-centric coordinate system by centering skeletal body and aligning it to transformed axis. Although pre-processing strategy can help to reduce view variation, but has drawbacks. Given that the skeleton is not rigid, the human-defined strategy may not be flexible enough to handle various real-time situations. However, these processes are not explicitly built to optimize action recognition in mind but are based on priori information, limiting the space for utilizing effective viewpoints. Developing a system that learns optimal viewpoints while mitigating the impact of viewpoint diversity without requiring much human effort remains an unresolved topic that merits further study.

We intend to address the view variation problem in this work to achieve efficiency for skeleton-based action recognition. We propose a view-adaptive (VA) scheme that automatically determines the observation viewpoint within the network architecture for each sequence, rather than pre-processing the 3D skeletons based on fixed human-defined criteria to reduce view variations. The classification stage gets skeleton in a new viewpoint that best suits efficient recognition. We designed an end-to-end view-adaptive graph neural network (VA-GNN) which consists of a view-adaptive module and GNN modules for accurate recognition (Fig. 2). The observed viewpoint is equivalent to the transformation of skeletal to new coordinate value adaptively. The learned view-adaptive viewpoints are input for the classification of human action sequences. The view-adaptive module and the classification network are trained to learn and determine the appropriate virtual viewpoints. We conducted experiments to demonstrate the efficacy of our proposed view-adaptive mechanism using the graph neural network (VA-GNN). The following three are the key contributions:

- We presented a view-adaptive (VA) mechanism to re-position the human skeleton viewpoints for improved recognition accuracy. Our method liberates the human resources spent on developing extensive pre-processing criteria to handle a wide range of situations.
- The designed view-adaptive strategy is embedded into an end-to-end GNN. Our VA-GNN architecture dynamically defines observation viewpoints for recognition during the training process.
- Extensive experiments on three large-scale action recognition datasets demonstrate the effectiveness of the VA-GNN. The impact of parameters and network components are investigated. Furthermore, we show that the gain is brought by our VA module rather than by simply increasing the number of GNN layers.

Fig. 2. A pipeline of the end-to-end view-adaptive graph neural network (VA-GNN). The architecture contains a view-adaptive network and graph neural network for classifying human actions. The VA subnetwork calculates the observed view dynamically and turns the original skeletal input into new viewpoints for the classification in the GNN block.

The remaining paper is organized as follows: Section II provides an overview of the related work regarding the proposed work. Section III and Section IV explains the proposed methodology in this research. Experimental results and their analysis are presented in Section V. The paper is concluded with future works in Section VI.

II. RELATED WORK

A. View Invariant Action Recognition

Human actions can be captured from any camera angles. As a result, developing and implementing action recognition techniques have become difficult [12]. Researchers spent a lot of time and effort developing view-invariant action recognition for RGB-based action recognition [13], [14], [15], [16], [17], [18], [27], [28], [29].

One category of the view invariant action recognition uses as many views as possible for training the model. Fuzzy distances from the human body posture are used to represent the invariant action recognition, and multiple cameras are used to design a Bayesian framework for various viewpoints [13]. In reality, it is costly to capture video sequences from a large number of views.

The second category is designing the view invariant feature description using the motion curvature. In [14], the view invariance is achieved by the articulated body motion from point triplets and similarity of transition between two subjects. Affine representations based on centroid and curvature are defined to get view-invariant motion trajectory [15]. Work in [16] also uses the spatiotemporal curvature of the motion trajectory to define a view-invariant description for action recognition. Usually, the transformed domain descriptors are prone to loss from the input videos.

Another category employs knowledge transfer using the source and target views, respectively. The virtual views are presented using knowledge transfer from source view to target view for action description [17]. In [18], a cross-view action recognition from the temporal self-similarities is presented. The appearance feature is used to learn activities from the wrong viewpoint [27]. The virtual view kernel is used to compute the similarity of the finite-dimensional feature obtained by linear transformation [28]. They use latent space for features to obtain different views, which requires a significant amount of human effort. However, these techniques for RGB videos cannot be directly extended for skeleton-based action recognition.

B. Viewpoint Skeleton-based Action Recognition

The skeleton-based action recognition is novel from a viewpoint perspective. For view variation, common used methods are based on pre-processing the input skeleton [2], [20], [21], [22], [24], [25], [26], [30], [31], [32]. The skeleton is transformed to a centered orientation with body alignment using pre-processing methods, which affect motion sequences. For example, the running action becomes running in same place, and the action of sitting on a chair becomes sitting on standing orientation. The pre-processing applied on skeleton sequences is the same for all frames regardless of motion in action category, making the motion sequence invariant to the fixed body orientation. However, the 3D rotation in pre-processing presented in [32], is fixed to the x -axis parallel to the “left” and “right shoulder and the y -axis towards the “spine” and “hip,” which is not always suitable for orientation alignment. In [20], the up-side direction of the pose in the sequence of frames by averaging the rotation transformation. However, the upside pose may not be feasible for all kinds of actions. For example, “picking object” with one hand from the ground.

The pre-defined criteria for the skeleton transformation are complex to handle the non-rigid human skeleton. Therefore, we employed view-adaptive module to determine and learn appropriate viewpoints for robust transformations. By doing so, the skeletons are transformed to the best representations under the views for each action sequence individually.

Fig. 3. Illustration of human body joints and bones to represent the skeleton information. (a) shows the twenty-five joint labels of the NTU-RGB-D-60 [22] and NTU-RGB-D-120 [31] datasets. (b) shows the eighteen points labels of the Kinetics-Skeleton-400 dataset [33]. The lines combining the joints/nodes represent the bone's structure as edges.

C. GNN for Skeleton-based Action Recognition

Graph neural networks have proven the effectiveness of the structured data and are used to model human skeleton data. The spatiotemporal graph convolution neural network is introduced by treating human joints as nodes and bones as edges of a graph. Several recent works [2], [6], [34], [35], [32], [20], [36], [37], [38], [39], attempt to convert 3D skeleton to graph and then leverage graph convolution neural networks which significantly outperformed previous approaches.

The human body joints as nodes and bones as edges denote the relationship for representing spatial graph as an adjacency matrix. This kind of graph is naturally defined based on the human body structure. The method presented in [2] enhanced the pre-defined natural graph by defining additional physical connections adaptively to joint pairs and edges for better graph construction. It also uses a multi-stream ensemble using joints and bone data to increase performance.

Disentangling and unifying GCN [20] used joint features at spatiotemporal neighborhoods for multiscaling. This method also pre-processed skeleton data for normalization and translation. Directed GNN [39], used standard deviation across each channel to get the skeleton energy, and translated the skeleton to the central perspective. A similar kind of approach is used in [22] and [31]. The 3D bio-Constrained skeleton method limits the joint motion and bone lengths [6]. They use a fixed angle of 150° from the base of the “spine-end” to “hip-left”, “hip-right,” and “spine-mid” for motion limits. However, the added range limit of joints motion with respect to the bone using Euler angles is set empirically. They also set criteria for discarding the skeleton, which may not be suitable for certain actions.

The majority of the primary research concentrates on mapping a skeleton sequence to an image or invariant feature descriptor. Still, such works ignore the constraints of view variations in the human skeletal sequences.

III. MODELING OF VIEW-ADAPTIVE MODULE

Even for the same action, the skeleton descriptions in different views are quite different. It would lead to even bigger intra-class differences than inter-class differences caused by view changes. This variation of view makes recognition of human activity complicated.

We present an end-to-end VA-GNN design to eliminate the effects of viewpoints variation, which dynamically re-observes skeletal in frames from different view perspectives before classification. Our proposed network consists of a view-adaptive (VA) module and an adaptive graph neural network (GNN). The VA block allows the network to learn the best-observed viewpoints and provide a transformed skeleton sequence in an unsupervised manner. Then, the transformed skeleton is converted to a graph to determine the discriminative feature by graph neural network block in an optimized end-to-end manner.

Specifically, we denote the skeleton joint as $S_{t,j}$ for j^{th} skeleton joint in the t^{th} frame as follow:

$$S_{t,j} = [x_{t,j}, y_{t,j}, z_{t,j}]^T \in \mathcal{Q}, \quad (1)$$

where $t \in (1, \dots, \mathcal{F})$, $j \in (1, \dots, J)$. $\mathcal{F}$ is the total number of frames, and J shows the total number of joints in the skeleton, which is determined by the skeleton estimation algorithm (see Fig. 3). $S_{t,j}$ contain three components, i.e., coordinate values x, y, z , at the time t for the joint j . We refer it to original coordinate space $\mathcal{Q}$ as: $S_{t,j} \in \mathcal{Q}$. The input skeleton sequence S is sensitive to view variations as illustrated in Fig. 2, we transformed them to $\tilde{S}_{t,j} = [\tilde{x}_{t,j}, \tilde{y}_{t,j}, \tilde{z}_{t,j}]^T$ by a view transformation expressed in VA block as follows:

$$\tilde{S}_{t,j} = \mathcal{T} S_{t,j}, \quad (2)$$

where $\mathcal{T} = (\mathcal{R}_x^{t,\alpha}, \mathbf{b}_t)(\mathcal{R}_y^{t,\beta}, \mathbf{b}_t)(\mathcal{R}_z^{t,\gamma}, \mathbf{b}_t)$ is the transformation function, $\mathcal{R}^t = [\mathcal{R}_x^{t,\alpha}, \mathcal{R}_y^{t,\beta}, \mathcal{R}_z^{t,\gamma}] \in \mathbb{R}^{3 \times 3}$ is rotation matrix and $\mathbf{b} \in \mathbb{R}^3$ is the translation vector. Let $\mathcal{R}_x^{t,\alpha}$ denote the rotation of original x -coordinate around the x -axis by α radians, as:

$$\mathcal{R}_x^{t,\alpha} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha_t & -\sin \alpha_t \\ 0 & \sin \alpha_t & \cos \alpha_t \end{bmatrix}. \quad (3)$$

Fig. 4. Illustration of proposed view-adaptive graph neural network (VA-GNN) architecture. The view-adaptive block is used at the beginning of method for optimal viewpoint learning. The building block of our GNN consist four residual blocks (AGCN) built by Adaptive Graph Convolution (*AGCConv*) layer with 128, 192, 192, 128 channels, respectively. The *AGCConv* layer is followed by 2D convolution layer, batch normalization and a ReLU activation function with same number of channels. Finally, the linear layer is used to obtain the classification score.

Similarly, the rotation matrix $\mathcal{R}_y^{t,\beta}$ and $\mathcal{R}_z^{t,\gamma}$ are defined as:

$$\mathcal{R}_y^{t,\beta} = \begin{bmatrix} \cos \beta_t & 0 & \sin \beta_t \\ 0 & 1 & 0 \\ -\sin \beta_t & 0 & \cos \beta_t \end{bmatrix}, \quad (4)$$

$$\mathcal{R}_z^{t,\gamma} = \begin{bmatrix} \cos \gamma_t & -\sin \gamma_t & 0 \\ \sin \gamma_t & \cos \gamma_t & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (5)$$

Note that all the skeleton joints in the t^{th} frame share the same parameters $(\alpha, \beta, \gamma, \mathbf{b})$ for transformation because of the change in the viewpoint is a rigid motion. The parameters α, β, γ are initialized by random value ranged from -5 to 5. So, combining the learned variations using parameters in values $\tilde{x}, \tilde{y}, \tilde{z}$ with t and j , the original space $\mathcal{Q}$ is transformed into a new space $\tilde{\mathcal{Q}}$. For different frames, the viewpoints can change across time. We found that our method can preserve more relative rotations among the original skeleton, which overcomes the problem of variation in the viewpoint of the skeleton during the training of the GNN dynamically.

IV. VIEW-ADAPTIVE GRAPH NEURAL NETWORK

The detailed architecture of the proposed view-adaptive graph neural network (VA-GNN) is illustrated in Fig. 4. In the first instance, after feeding the video data, the VA module transforms the skeleton to new representations under suitably observed viewpoints. Afterwards, the GNN block is for recognizing the actions from transformed skeletons. In the end, the learned features are classified using the traditional linear layer. VA-GNN is trained end-to-end by minimizing the loss and optimizing the human action recognition performance. The view-adaptive formulation, graph convolutions, adaptive graph convolution and overall architecture are presented in the following sub-sections.

A. View-Adaptive Formulation

The view-adaptive module automatically learn and determine the observation viewpoints with the transformation parameters $\alpha_t, \beta_t, \gamma_t, \mathbf{b}_t$, as described in Section 3. The GNN module learns the temporal dynamics from the adopted skeleton data to recognize the human action in an end-to-end manner. The observation can be considered skeleton's repositing, which is characterized by rotation and translation. The obtained individual t^{th} frame from the sequence, with the skeleton $\tilde{\mathcal{S}}_{t,j}$ as input to the graph neural network module are utilized to learn the rotation key parameters $\alpha_t, \beta_t, \gamma_t$ to

obtain the rotation matrix $\mathcal{R}^t$, and the translation vector $\mathbf{b}_t$. These parameters are learned by using the linear layer with an unsupervised manner (i.e., without direct supervision for $\alpha_t, \beta_t, \gamma_t, \mathbf{b}_t$). For the parameters of the rotation matrices, they are learned as:

$$[\alpha_t, \beta_t, \gamma_t]^T = \mathbf{W}_r \mathbf{k}_t^r + \mathbf{c}_r, \quad (6)$$

here $\mathbf{W}_r \in \mathbb{R}^{3 \times n}$ denotes the weight matrix, $\mathbf{k}_t^r \in \mathbb{R}^{n \times 1}$ is a vector of linear layer of output with n number of neurons and $\mathbf{c}_r \in \mathbb{R}^{3 \times 1}$ is an offset (a bias) vector of linear layer. With the parameters learned, the $\mathcal{R}^t$ is obtained using equation 2. The part of translation parameters also consist of linear layer and calculated as:

$$\mathbf{b}_t = \mathbf{W}_b \mathbf{k}_t^b + \mathbf{c}_b, \quad (7)$$

here $\mathbf{W}_b \in \mathbb{R}^{3 \times n}$ denotes the weight matrix, $\mathbf{k}_t^b \in \mathbb{R}^{n \times 1}$ is the output vector of the layer and $\mathbf{c}_b \in \mathbb{R}^{3 \times 1}$ is the offset vector.

B. Graph Convolutions

The best-suited viewpoint skeleton for the sequences $\tilde{\mathcal{S}}_{t,j}$ is obtained from the view-adaptive block. The adopted human skeleton $\tilde{\mathcal{S}}_{t,j}$ is then converted into graph representation: where $\mathcal{N} = \{n_1, \dots, n_J\}$ is the set joints as total J nodes, and $\mathcal{E}$ represents edges, which is the connectivity of joints. The l^{th} graph convolution applied to J^{th} node in two steps: (1) the node representations are transformed to learnable parameter matrix $\mathbf{W}_{\mathcal{G}_S} \in \mathbb{R}^{(l+1) \times l}$. (2) these transformed nodes are collected to n_j node with respect to the neighboring nodes and followed by non-linear ReLU function (σ). Let node representations are contained in the matrix $\mathcal{X}^l \in \mathbb{R}^{l \times J}$, then convolution operation is expressed as:

$$\mathcal{X}^{(l+1)} = \sigma(\mathbf{W}_{\mathcal{G}_S} \mathcal{X}^{(l)} \mathbf{A}'), \quad (8)$$

where $\mathbf{A}' \in [0, 1]^{J \times J}$ is the symmetric adjacency matrix of graph $\mathcal{G}_S$. The representation of skeleton data as graph adjacency $\mathbf{A}'$ is normalized using regularize sparse matrix [40] as:

$$\mathbf{A}' = (\mathbf{D}_\tau)^{-1/2} \mathbf{A} (\mathbf{D}_\tau)^{-1/2}, \quad (9)$$

here $\mathbf{A}$ is the basic adjacency matrix, $\mathbf{D}_\tau = \mathbf{D} + \tau \mathbf{I}$ in which $\mathbf{D}$ represents the degree matrix with constant τ and $\mathbf{I}$ represents the identity matrix.

C. Adaptive Graph Convolution

Fig. 5. Visualization of adjacency matrix in the AGCN block. At different number of epochs, AGCN block learnt a different weights for the graph in the network during the training process, while the x -axis and y -axis represents the nodes on the human body as: 0. Pelvis, 1. Thorax, 2. Neck, 3. Head, 4. L-Shoulder, 5. L-Elbow, 6. L-Wrist, 7. L-Hand, 8. R-Shoulder, 9. R-Elbow, 10. R-Wrist, 11. R-Hand, 12. L-Hip, 13. L-Knee, 14. L-Ankle, 15. L-Feet, 16. R-Hip, 17. R-Knee, 18. R-Ankle, 19. R-Feet, 20. Spine, 21. L-Hand Fingers, 22. L-Thumb, 23. R-Hand Fingers, 24. R-Thumb. Best viewed in colour.

To improve the flexibility of the graph, we also added a learnable weight matrix $\mathbf{M} \in \mathbb{R}^{J \times J}$ to conventional graph convolution to learn a unique topology for each sample. So, equation 8 is transformed to:

$$\mathbf{X}^{(l+1)} = \sigma(\mathbf{W}_{\mathcal{G}_S} \mathbf{X}^{(l)} \mathbf{A}' \odot \mathbf{M}). \quad (10)$$

The second element $\mathbf{M}$ is also $J \times J$ adjacency matrix. In comparison to $\mathbf{A}'$, in the training process, the $\mathbf{M}$ factor is parametrized and optimized along with other parameters. There are no limitations on the value of $\mathbf{M}$, which means that the graph is fully learned by input training data ($\tilde{\mathcal{S}}_{t,j}$). The model learns graphs that are entirely targeted for action recognition in this data-driven manner. Note that the element may be an arbitrary value in the matrix. It implies the significance of relations between two joints and the strength of the connections by adding edges adaptively. This adaptive graph convolution process is inspired from [2], [39] in addition to the view-adaptive strategy.

D. Overall Architecture

The final model architecture contains the view-adaptive block to observe the human body skeleton from best-viewed variations within the sequences, followed by graph convolution blocks. In the end, the linear layer is used to map the score for the classifier. The view-adaptive block is used to handle the variation in viewpoints. Then the best-observed viewpoint skeletal is followed to the GNN block. VA block is only used at the beginning of the network as a stem block. Our graph neural network consists of four residual blocks [41], which is constructed by the Adaptive Graph Convolution (AGConv) layer [2] as a baseline. The graph convolution blocks handle the spatiotemporal data with 128, 192, 192, and 128 feature channels, respectively, followed by batch normalization [42] and a ReLU [43]. The AGCN block is repeated four times and followed by a 2D convolution layer, batch normalization, and a ReLU activation function. All graph convolution blocks, except the first block, downsample the temporal dimension with stride two in temporal convolution operation by *Conv 2D layer* and sliding window. Five residuals connections between the layers are added to facilitate the training and prevent overfitting in our method [41]. Various experiments are performed to evaluate the performance of our proposed strategy in the following section.

V. EXPERIMENTAL RESULTS

We evaluated the effectiveness of our proposed VA-GNN framework on three benchmark datasets, which includes NTU RGB+D-60 (NTU60) [22], NTU RGB+D-120 (NTU120) [31], and Kinetics-Skeleton-400 (KS400) [33]. The specific details of these datasets are described in Table I. The details of training and componential study are described in this section. The final model is then evaluated and compared with state-of-the-art FER approaches.

A. Datasets

1) *NTU60*: This dataset was captured using the Kinect sensor, which is currently the largest dataset with RGB and depth video and skeleton data for human action recognition. It contains 60 human action categories that reflect daily actions, medical conditions, two-person interactions with joint actions. Each human skeleton has 25 joints position on important locations on the body part. The different camera setups, viewpoints, angle orientations of the subjects result in various samples. The two standard protocols, Cross View (CV) and Cross Subject (CS) are used to collect the dataset. For CV, three camera positions are used to capture the samples, where “camera 2” and “camera 3” are used for the training and “camera 1” for the testing set, and CS, where 40 subjects are split into training and testing groups. The difference in camera views, multiple subjects, and video samples makes this dataset challenging for action recognition.

2) *NTU120*: This dataset extends NTU60 with an additional 60 action classes. The 106 subjects are captured using 32 different camera setups. By adding 57,600 videos, the total number of skeleton sequences are increased to 114,480 samples. Two different protocols are used for evaluation purpose, i.e., first, CV is replaced by Cross Setup (CX), and the other is Cross Subject (CS). In the CX protocol, 55,003 video samples from half of the camera settings are used for training, and 59,477 video samples from the rest of half camera setup are selected for testing. In CS protocol, 63,026 video samples from 53 subjects are used for training purpose, and the rest of 51,454 videos from 67 subjects are kept for testing.

3) *KS400*: Kinetics Skeleton dataset is obtained from the kinetics video action recognition dataset by employing the OpenPose [44] human pose estimation. This dataset contains 400 different action classes with 240,436 training and 19,796 testing sequences. Each skeleton contains 2D spatial

coordinate position of 18 body points along with the confidence score. The skeletons with lower confidence levels are discarded. The maximum two skeletons are kept in the video sequences. Due to the limitation of not having third dimension information for this dataset, we employed homogeneous coordinates representation as $w = 1$ to add a third coordinate to every 2D point.

TABLE I
CHARACTERISTICS OF EXPERIMENTAL DATA

Dataset	Subject	No. of Markers	Videos	Action
NTU60	40	25	56,880	60
NTU120	106	25	114,480	120
KS400	-	18	260,232	400

B. Training Details

The experiments are performed using the deep convolutional neural network architecture in the PyTorch library. The training and validation computations are performed on two Nvidia GTX 1080 graphic processing units, 64 gigabytes of ram, and Intel Core i7-7800x CPU at the rate of 3.50GHz 12x processors. As an optimization strategy, models are trained using a stochastic gradient with a momentum of 0.9. The batch size is 16 for training and 32 for testing, with cross-entropy as the loss function. 0.0005 is the weight decay rate. A batch size of 128 is used for training, and 256 is used for testing in KS400 dataset.

C. Componental Study

In the final architecture of the proposed VA-GNN, we examine the various elements and their configurations. For in-depth analysis and study, results are reported as classification accuracy using the NTU60 dataset.

TABLE II
COMPARISON OF PROPOSED VIEW-ADAPTIVE BLOCK AND PRE-PROCESSING METHOD ON THE NTU60 DATASET

	Methods	CS (%)	CV (%)
w/o VA	VA-GNN*	76.11	89.57
	G3D-P	81.15	92.08
	A-GCN-P	84.30	92.70
	VA-GNN-P	80.88	92.31
w/o Pre-proc.	G3D	74.57	86.89
	A-GCN	82.40	90.11
	VA-GNN	86.21	94.18

1) *Comparison with Pre-processing Strategies*: Some approaches pre-process the skeletons using human-defined criteria to overcome the problems created by view variations [2], [19], [20], [21], [22], [23], [24], [25], [26]. As specified in section IV, our proposed VA block is only used at the beginning of the network as a stem block. We tested the performance of our model with the VA block, pre-processing (P) method, and without VA block. The summary for this study is described in Table II. For a fair comparison, we also implemented the source code of two state-of-the-art methods, A-GCN [32] and G3D [20], and tested the performance with and without pre-processing method. Our model VA-GNN-P with pre-processing method improved accuracy of +2.74% from the baseline model VA-GNN without VA block using CV protocol. The test with view adaption block and the baseline network (VA-GNN) improved +1.87% accuracy in CV protocol as compare to the VA-GNN-P model. Using the

same pre-processing method used as in A-GCN [32], we applied the pre-processing. As shown in Table I, using the CV data protocol and stated pre-processing techniques, our proposed VA-GNN outperforms A-GCN-P by +1.07% and G3D-P by +2.10%. The pre-processing strategy defined by humans is not optimum for action recognition. Because the human body is non-rigid, defining rotation criteria is not always appropriate for the purpose of orientation alignment. Our method uses a network that has been trained by optimizing classification accuracy to find the appropriate viewpoints automatically.

TABLE III
COMPARISON ON VIEW-ADAPTIVE LAYER AND DIFFERENT NUMBER OF GCN BLOCKS ON THE NTU60 DATASET

Model	Module	No. Params.(M)	CS (%)	CV (%)
VA-GNN	VA	0.31	78.50	81.03
	1 AGC	0.63	79.22	84.33
	2 AGC	1.30	83.69	88.10
	3 AGC	1.97	85.11	91.83
	4 AGC	2.33	87.04	93.61
	5 AGC	2.98	86.48	89.27
	6 AGC	3.61	88.20	92.11

2) *Influence of the Model Parameters*: The view-adaptive module is part of our proposed recognition system and it contains a fewer number of parameters than the GCN part. There are two ways to build up the model over the number of network parameters for the recognition. (1) Stacking layers in GCN blocks and (2) increasing the number of neurons through the depth of layers in GCN block. Note we had used A-GCN as our baseline network. So, the original proposed A-GCN contains ten numbers of GCN blocks. Therefore, we show the results when changing the number of layers in GCN blocks.

Table III shows the comparison between the GCN modules and our proposed VA module with a different number of stacking layers/blocks. The depth of GCN blocks is 128, 192, 192, and 128 in width, impacting the performance by the number of parameters, although it has been fixed for consistency after trials during experiments. Stacking more layers or utilizing a greater number of neurons to increase parameters is not as efficient as our proposed VA module. As compared to baseline models, our model outperforms with less number of parameters.

3) *Visualization of the Learned Adjacency Matrix*: The adjacency matrix during the optimization is visualized in Fig. 5. The first figure shows the epoch 10 graph structure, which is very similar to the natural connectivity of the human body and the self-connections have the highest weights. After several epochs of training, it can be seen that the adjacency matrix is adaptively changed by the AGCN block. This learned adjacency matrix would be much better than the physically defined adjacency matrix, because it adaptively changes the weights to represent the relationship of the learned features of nodes.

D. Comparison to other Approaches

In this section, we presented the performance comparison between our proposed and other state-of-the-art methods. The comparison not only includes GNN based action recognition based methods, but also CNN and RNN based ones.

1) *NTU60 Dataset*: We used the standard CS and CV protocol presented in [22] for assessing the performance. The

results are presented in Table IV. This dataset contains a lot of variations in the view, which makes action recognition difficult. Compared with CNN-based methods, all GNN used methods have a much smaller model size and a slightly higher or similar accuracy. Most RNN based methods do not perform well on the NTU60 dataset. In the GNN category, our VA-GNN outperforms RA-GCN by 0.4% and 1.1% on CS and CV settings. Although A-GCN performs slightly better than our method, the number of parameters of that is almost twice as our VA-GNN.

TABLE IV
COMPARISON WITH EXISTING METHODS ON NTU-RGB-D-60 USING CROSS SUBJECT (CS) AND CROSS VIEW (CV) PROTOCOL

Features	Methods	Year	No. Parns. (M)	NTU60 Accuracy		
				Data	CS (%)	CV (%)
CNN	VI-CNN [45]	2017	-	J.Skel.	88.0	87.2
	JEAs [6]	2019	-	J.Skel.	75.9	81.8
	FI-CNN [11]	2021	22.4	J.Skel.	84.2	89.7
RNN	PLSTM [22]	2017	-	J.Skel.	62.9	70.3
	LSTMTG [21]	2018	-	J.Skel.	69.2	77.7
	P&C [46]	2020	-	J.Skel.	50.7	76.1
GNN	P-GCN [47]	2018	-	J.Skel.	87.5	93.2
	STGR [48]	2018	-	J.Skel.	86.3	74.9
	A-GCN [2]	2019	3.8	J.Skel.	88.0	95.1
	SGN [49]	2020	0.7	J.Skel.	86.9	92.8
	HGNN[37]	2021	-	J.Skel.	-	93.7
	RA-GCN [50]	2021	2.0	J.Skel.	85.8	93.1
	VA-GNN (ours)	-	2.3	J.Skel.	86.2	94.2

2) *NTU-RGB-D-120 Dataset*: We used the standard CS and CX protocol presented in [31], and the results are presented in Table V. The dataset adds 60 more fine-grained action categories, which makes the task more challenging. Similarly, all GNN based methods perform significantly better than CNN and RNN based methods. Compared with FI-CNN [11], our VA-GNN improved 4.3% in CS and 4.4% in CX. Among all GNN based methods using single-joint skeleton information/stream, our GNN achieved the state-of-the-art result, suppressing SGN [49] 1.7% in CS and 2.1% in CX, and suppressing RA-GCN [50] by 0.9% in CS and 1.3% in CX. Although MS-G3D [20] is slightly better than ours, it utilizes two-stream information, which is not used in our method.

TABLE V
COMPARISON WITH EXISTING METHODS ON NTU-RGB-D-120 USING CROSS SUBJECT (CS) AND CROSS SETUP (CX) PROTOCOL

Features	Methods	Year	No. Parns. (M)	NTU120 Accuracy		
				Data	CS (%)	CX (%)
CNN	RCMT [51]	2018	-	J.Skel.	62.2	61.8
	FI-CNN [11]	2021	22.4	J.Skel.	74.8	76.9
RNN	PLSTM [22]	2017	-	J.Skel.	25.5	26.3
	LSTMTG [21]	2018	-	J.Skel.	55.7	57.9
GNN	MS-G3D [20]	2020	3.2	2s.Skel.	86.9	88.4
	SGN [49]	2020	0.7	J.Skel.	77.4	79.2
	RA-GCN [50]	2021	2.1	J.Skel.	78.2	80.0
	VA-GNN (ours)	-	2.6	J.Skel.	79.1	81.3

3) *Kinetics Skeleton 400 Dataset*: Our VA-GNN achieved 32.9% top-1 accuracy and 57.5% top-5 accuracy. The summary of results is presented in Table VI. Among all methods using single joint skeleton information, our VA-GNN is higher than AS-GCN [38] by 1.0% and STGR [48] by 1.4% in terms of top-5 accuracy. Although A-GCN [2] and HGNN [37] perform slightly better than us, these methods use two or three streams by leveraging additional information than our method. However, these multi-stream methods [2], [37] lead to a significant increase in terms of the training and

processing time. Our VA-GNN does not include additional information and would be much light and faster in the real-time applications.

TABLE VI
COMPARISON WITH EXISTING METHODS KINETICS SKELETON 400 USING TOP 1 AND TOP 5 ACCURACIES

Features	Methods	Year	KS400 Accuracy		
			Data	Top 1 (%)	Top 5 (%)
GCN	AS-GCN [38]	2019	J.Skel.	34.8	56.5
	HGNN [37]	2021	3s-Skel.	37.1	60.0
	A-GCN [2]	2019	2s-Skel.	36.1	58.7
	STGR [48]	2018	J.Skel.	33.6	56.1
	VA-GNN (ours)	-	J.Skel.	32.9	57.5

VI. CONCLUSION AND FUTURE WORK

This paper proposed a method for view-adaptive of 3D skeleton-based action recognition on an adaptive graph convolution neural network model. The presented VA-GNN model is capable of changing the observation viewpoint to the appropriate ones through an unsupervised module instead of human pre-defined criteria for re-positioning the skeletons for action recognition. The goal of this optimization in the VA module is to maximize recognition performance. We have designed a view-adaptive mechanism based on the graph neural network. The VA module transforms the variations in the human body skeleton's viewpoints and eliminates the influence of diversity in views from camera positions. Besides the VA block, an adaptive graph neural network module eases the training and allows end-to-end learning of best graph topology based on the input video sequences. The experimental results show that the proposed framework architecture contributes recognition performance on leading datasets and achieved state-of-the-art performances with reduced parameters, which is feasible for implementation in future works along with the pose estimation in an end-to-end manner.

REFERENCES

- [1] R. Poppe, 'A survey on vision-based human action recognition', *Image Vis. Comput.*, vol. 28, no. 6, pp. 976–990, 2010.
- [2] L. Shi, Y. Zhang, J. Cheng, and H. Lu, 'Skeleton-Based Action Recognition With Multi-Stream Adaptive Graph Convolutional Networks', *IEEE Trans. Image Process.*, vol. 29, pp. 9532–9545, Oct. 2020.
- [3] L. Lo Presti and M. La Cascia, '3D skeleton-based human action classification: A survey', *Pattern Recognit.*, vol. 53, pp. 130–147, 2016.
- [4] W. Gong *et al.*, 'Human Pose Estimation from Monocular Images: A Comprehensive Survey', *Sensors*, vol. 16, no. 12, p. 1966, Nov. 2016.
- [5] F. Han, B. Reily, W. Hoff, and H. Zhang, 'Space-time representation of people based on 3D skeletal data: A review', *Comput. Vis. Image Underst.*, vol. 158, pp. 85–105, 2017.
- [6] Q. Nie, J. Wang, X. Wang, and Y. Liu, 'View-Invariant Human Action Recognition Based on a 3D Bio-Constrained Skeleton Model', *IEEE Trans. Image Process.*, vol. 28, no. 8, pp. 3959–3972, 2019.
- [7] 'Stereolabs - Capture the World in 3D'. [Online]. Available: <https://www.stereolabs.com/>. [Accessed:

- 31-May-2021].
- [8] 'Intel® RealSense™ Computer Vision - Depth and Tracking cameras'. [Online]. Available: <https://www.intelrealsense.com/>. [Accessed: 31-May-2021].
 - [9] Z. Zhang, 'Microsoft kinect sensor and its effect', *IEEE Multimedia*. 2012.
 - [10] B. Xiao, H. Wu, and Y. Wei, 'Simple baselines for human pose estimation and tracking', in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 466–481.
 - [11] A. Banerjee, P. K. Singh, and R. Sarkar, 'Fuzzy Integral-Based CNN Classifier Fusion for 3D Skeleton Action Recognition', *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 6, pp. 2206–2216, Jun. 2021.
 - [12] Xiaofei Ji and Honghai Liu, 'Advances in View-Invariant Human Motion Analysis: A Review', *IEEE Trans. Syst. Man, Cybern. Part C (Applications Rev.)*, vol. 40, no. 1, pp. 13–24, Jan. 2010.
 - [13] A. Iosifidis, A. Tefas, and I. Pitas, 'View-invariant action recognition based on artificial neural networks', *IEEE Trans. Neural Networks Learn. Syst.*, vol. 23, no. 3, pp. 412–424, 2012.
 - [14] Y. Shen and H. Foroosh, 'View-invariant action recognition from point triplets', *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 10, pp. 1898–1905, 2009.
 - [15] F. I. Bashir, A. A. Khokhar, D. Schonfeld, F. I. Bashir, A. A. Khokhar, and D. Schonfeld, 'View-invariant motion trajectory-based activity classification and recognition', *Multimed. Syst.*, vol. 12, no. 1, pp. 45–54, 2006.
 - [16] C. Rao and M. Shah, 'View-invariance in action recognition', in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001, vol. 2.
 - [17] R. Li and T. Zickler, 'Discriminative virtual views for cross-view action recognition', in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2855–2862.
 - [18] I. N. Junejo, E. Dexter, I. Laptev, and P. Pérez, 'Cross-View Action Recognition from Temporal Self-similarities', in *Computer Vision -- ECCV 2008*, 2008, vol. 5303 LNCS, no. PART 2, pp. 293–306.
 - [19] J. Weng, C. Weng, J. Yuan, and Z. Liu, 'Discriminative spatio-temporal pattern discovery for 3D action recognition', *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 4, pp. 1077–1089, 2019.
 - [20] Z. Liu, H. Zhang, Z. Chen, Z. Wang, and W. Ouyang, 'Disentangling and Unifying Graph Convolutions for Skeleton-Based Action Recognition', in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 140–149.
 - [21] J. Liu, A. Shahroudy, D. Xu, A. C. Kot, and G. Wang, 'Skeleton-Based Action Recognition Using Spatio-Temporal LSTM Network with Trust Gates', *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 3007–3021, 2018.
 - [22] A. Shahroudy *et al.*, 'NTU RGB+D: A large scale dataset for 3D human activity analysis', in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2017, vol. 2016-Decem, no. 1, pp. 1010–1019.
 - [23] W. Zhu *et al.*, 'Co-Occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks', in *30th AAAI Conference on Artificial Intelligence, AAAI 2016*, 2016, vol. 30, no. 1, pp. 3697–3703.
 - [24] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu, 'An End-to-End Spatio-Temporal Attention Model for Human Action Recognition from Skeleton Data', Feb. 2017.
 - [25] M. Jiang, J. Kong, G. Bebis, and H. Huo, 'Informative joints based human action recognition using skeleton contexts', *Signal Process. Image Commun.*, vol. 33, pp. 29–40, Apr. 2015.
 - [26] Y. Du, W. Wang, and L. Wang, 'Hierarchical recurrent neural network for skeleton based action recognition', in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015, vol. 07-12-June, pp. 1110–1118.
 - [27] A. Farhadi and M. K. Tabrizi, 'Learning to Recognize Activities from the Wrong View Point', in *Computer Vision -- ECCV 2008*, 2008, vol. 5302 LNCS, no. PART 1, pp. 154–166.
 - [28] Z. Zhang, C. Wang, B. Xiao, W. Zhou, S. Liu, and C. Shi, 'Cross-view action recognition via a continuous virtual path', in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2690–2697.
 - [29] X. Wu and Y. Jia, 'View-invariant action recognition using latent kernelized structural SVM', in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2012, vol. 7576 LNCS, no. PART 5, pp. 411–424.
 - [30] J. Liu, G. Wang, L. Y. Duan, K. Abdiyeva, and A. C. Kot, 'Skeleton-Based Human Action Recognition with Global Context-Aware Attention LSTM Networks', *IEEE Trans. Image Process.*, vol. 27, no. 4, pp. 1586–1599, 2018.
 - [31] J. Liu, A. Shahroudy, M. Perez, G. Wang, L. Y. Duan, and A. C. Kot, 'NTU RGB+D 120: A Large-Scale Benchmark for 3D Human Activity Understanding', *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2684–2701, Oct. 2020.
 - [32] L. Shi, Y. Zhang, J. Cheng, and H. Lu, 'Skeleton-Based Action Recognition with Multi-Stream Adaptive Graph Convolutional Networks', *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2019-June, pp. 12018–12027, Dec. 2019.
 - [33] W. Kay *et al.*, 'The Kinetics Human Action Video Dataset', *arXiv*, May 2017.
 - [34] C. Li, Z. Cui, W. Zheng, C. Xu, and J. Yang, 'Spatio-temporal graph convolution for skeleton based action recognition', in *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, 2018.
 - [35] L. Zhao, X. Peng, Y. Tian, M. Kapadia, and Di. N. Metaxas, 'Semantic graph convolutional networks for 3D human pose regression', *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2019-June, pp. 3420–3430, 2019.
 - [36] T. Ahmad, H. Mao, L. Lin, and G. Tang, 'Action Recognition Using Attention-Joints Graph

Convolutional Neural Networks’, *IEEE Access*, vol. 8, pp. 305–313, 2020.

- [37] X. Hao, J. Li, Y. Guo, T. Jiang, and M. Yu, ‘Hypergraph Neural Network for Skeleton-based Action Recognition’, *IEEE Trans. Image Process.*, p. 1, 2021.
- [38] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, and Q. Tian, ‘Actional-structural graph convolutional networks for skeleton-based action recognition’, *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2019-June, pp. 3590–3598, 2019.
- [39] L. Shi, Y. Zhang, J. Cheng, and H. Lu, ‘Skeleton-based action recognition with directed graph neural networks’, *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2019-June, pp. 7904–7913, 2019.
- [40] A. Falah Yaqoob and B. Al-Sarray, ‘Enhancing Sparse Adjacency Matrix for Community Detection in Large Networks’, *Al-Nahrain J. Sci.*, vol. 22, no. 4, pp. 75–85, Dec. 2019.
- [41] K. He, X. Zhang, S. Ren, and J. Sun, ‘Deep residual learning for image recognition’, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [42] S. Ioffe and C. Szegedy, ‘Batch normalization: Accelerating deep network training by reducing internal covariate shift’, in *International conference on machine learning*, 2015, pp. 448–456.
- [43] V. Nair and G. E. Hinton, ‘Rectified linear units improve restricted boltzmann machines’, in *ICML*, 2010.
- [44] Z. Cao, T. Simon, S. E. Wei, and Y. Sheikh, ‘Realtime multi-person 2D pose estimation using part affinity fields’, in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017.
- [45] M. Liu, H. Liu, and C. Chen, ‘Enhanced skeleton visualization for view invariant human action recognition’, *Pattern Recognit.*, 2017.
- [46] K. Su, X. Liu, and E. Shlizerman, ‘PREDICT & CLUSTER: Unsupervised Skeleton Based Action Recognition’, pp. 9628–9637, 2020.
- [47] K. Thakkar and P. J. Narayanan, ‘Part-based graph convolutional network for action recognition’, in *Proceedings of the British Machine Vision Conference (BMVC)*, 2018, p. 270.
- [48] B. Li, X. Li, Z. Zhang, and F. Wu, ‘Spatio-temporal graph routing for skeleton-based action recognition’, in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019, vol. 33, no. 01, pp. 8561–8568.
- [49] P. Zhang, C. Lan, W. Zeng, J. Xing, J. Xue, and N. Zheng, ‘Semantics-Guided Neural Networks for Efficient Skeleton-Based Human Action Recognition’, in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 1109–1118.
- [50] Y.-F. Song, Z. Zhang, C. Shan, and L. Wang, ‘Richly activated graph convolutional network for robust skeleton-based action recognition’, *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 5, pp. 1915–1925, 2020.
- [51] Q. Ke, M. Bennamoun, S. An, F. Soheli, and F. Boussaid, ‘Learning Clip Representations for Skeleton-Based 3D Action Recognition’, *IEEE*

Trans. Image Process., vol. 27, no. 6, pp. 2842–2855, 2018.

recognition, deep learning and medical image processing.

research interests include graph matching, image segmentation, shape matching, and pattern recognition.

interests include computer vision, machine learning and deep learning.

interests include image processing, pattern recognition, and bioinformatics. He has over 600 journal and conference publications in these areas. Professor Yan is an IEEE Fellow and IAPR Fellow, and he received the 2016 Norbert Wiener Award from the IEEE SMC Society for contributions to image and biomolecular pattern recognition techniques. He is a member of the European Academy of Sciences and Arts.

Ali Raza Shahid received his master’s degree in electrical engineering from COMSATS University Islamabad, Pakistan, in 2015. He is currently pursuing the Ph.D. degree in Electrical Engineering at City University of Hong Kong. His research interests include pattern

Mehmood Nawaz received Ph.D. degree from City University of Hong Kong, Hong Kong SAR in 2020 and M.Sc. degree from Shanghai Jiaotong University, China in 2017. He is currently a Post Doc-Fellow in the Department of Electrical Engineering, Chinese University of Hong Kong. His

Xinqi Fan received his bachelor’s degree in automation from Southwest University, and master’s degree in electrical and electronic engineering from the University of Western Australia. He is currently pursuing his Ph.D. degree at the City University of Hong Kong. His research

Hong Yan received his Ph.D. degree from Yale University. He was Professor of Imaging Science at the University of Sydney and is currently Chair Professor of Computer Engineering and Wong Chung Hong Professor of Data Engineering at City University of Hong Kong. His research