Automatically human action recognition (HAR) with view variation from skeleton means of adaptive transformer network

Mehmood Faisal, Chen Enqing, Abbas Touqeer, Akbar Muhammad Azeem, Khan Arif Ali

DOI: 10.1007/s00500-023-08008-z

Automatically Human Action Recognition (HAR) with View Variation from Skeleton Means of Adaptive Transformer Network

Faisal Mehmood[1], Enqing Chen[1,2*†], Touqeer Abbas[3†], Muhammad Azeem Akbar[4†] and Arif Ali Khan[5†]

[1]School of Electrical and Information Engineering, Zhengzhou University, Zhengzhou, 450001, Henan, China.
[2]Henan Xintong Intelligent IOT Co., Ltd., No.1-303,Henan University, Zhengzhou, 450007, Henan, China.
[3]Department of Computer Science and Technology, Beijing University of Chemical Technology, Beijing, 100029, China.
[4]Department of Software Engineering, LUT University, Lahti, 15100, Finland.
[5]M3S Empirical Software Engineering Research Unit,University of Oulu, Oulu, 90570, Finland.

*Corresponding author(s). E-mail(s): ieeqchen@zzu.edu.cn;
Contributing authors: faisalmehmood685@uaf.edu.pk; 2017-uam-706@mnsuam.edu.pk; azeem.akbar@lut.fi; arif.khan@oulu.fi;
†These authors contributed equally to this work.

Abstract

Context: Human Action Recognition (HAR) using skeletons has become increasingly appealing to a growing number of researchers in recent years. ***Research Problem:*** It is particularly challenging to recognize actions when they are captured from different angles because there are so many variations in their representations. ***Objective:*** This paper proposes an automatic strategy for determining virtual observation viewpoints that are based on learning and data-driven to solve the problem of view variation throughout an act. ***Method:*** Our VA-CNN and VA-RNN networks, which use convolutional and recurrent neural networks with Long Short-term Memory (LSTM), offer an alternative to the conventional method of reorienting skeletons according to a human-defined earlier benchmark. ***Results:*** Using the unique view adaption module, each network first identifies the best observation perspectives and then transforms the skeletons for end-to-end detection with the main classification network based on those viewpoints. The suggested view adaptive models can provide significantly more consistent virtual viewpoints using the skeletons of different perspectives. By removing views, the models allow networks to learn action-specific properties more efficiently. Furthermore, we developed a two-stream scheme (referred to as VA-fusion) that integrates the performance of two networks to obtain an improved prediction. Random rotation of skeletal sequences is used to avoid overfitting during training and improve the reliability of view adaption models. ***Conclusion:*** An extensive experiment demonstrates that our proposed view-adaptive networks outperform existing solutions on five challenging benchmarks.

Keywords: Human Action, Skeleton, VA-CNN, VA-RNN, View adaptation

1 Introduction

A significant research topic in computer vision is human action recognition which has been studied a lot and made a lot of progress in the last few decades. In addition to visual surveillance, it can be used for human-computer interaction and video classification. It also controls games and video summaries and understands videos [1, 2]. Human action recognition is categorized into two types: 3D skeleton-based methods and 2D skeleton-based approaches, based on the input data. HAR based on RGB color has been extensively researched. Using 3D skeletons to show human's bodies has been getting more attention recently. This is because the locations of key joints in a human body are shown in 3D space. This study demonstrated that techniques for human action recognition founded on RGB and skeletal data complement one another [3, 4]. Skeletons have the advantage of being resistant to appearances, diversions, and other points of view as high-level visualizations [5, 6]. It has been discovered by researchers that even when no appearance information is provided, individuals can identify movements based on the movement of only a rare joint in the human body. With the widespread availability of affordable depth cameras [7], as in the Microsoft Kinect [8] and Intel RealSense [9], as well as the development of strong methodologies for human posture approximation from deepness [10], the achievement of 3D skeleton data has become straightforward. Following in the footsteps of many prior studies, including those reported in the survey publication [6, 11].

This work focuses on action recognition using skeletons. HAR is one of the most difficult problems to solve because of the large variety of viewpoints represented by the data collected from human actions. Large view variations are caused by two fundamental factors. First, in a real setting, the camera perspectives are adjustable, and changing camera perspectives result in major changes in skeletal representations, even when the sight is the same as before. Second, the actor could accomplish an action in many different ways. The orientations of this person may also change with time. When taken from several views, the skeleton models of the same position are quite different, as shown in Figure 1. In training, the variety of observation perspectives is extremely difficult the action recognition [5, 12]. It is possible that the perspectives of the testing samples were never observed from the viewpoints of the training samples, resulting in a considerable degradation in the recognition performance. Furthermore, in contrast to reliable opinions, a larger model is usually required to manage varied viewpoints. However, when it comes to training a larger model, it is more challenging to do so. In past studies, several attempts have been made to tackle the problem of perspective variance to attain robust action recognition [13-15]. Though, most of these workings are projected for usage in RGB-based action recognition systems [16]. Pre-processing management is regularly used to make the skeleton data invariant to the particular position and alignment of the body [17, 18]. The original 3D coordinates are translated into representations by engaging the body axis at the source and supporting the body level of the skeleton to be analogous to the (x, y) plane, using a person-centric coordinate system. A pre-processing method like this can somewhat reduce the difficulty of perspective fluctuation. However, it comes with some disadvantages as well. Given that the human body is not rigid, it is possible that the strategy specified by humans will not be flexible enough to deal with various situations. Because they depend on past information rather than on explicit design to enhance action recognition, these processing approaches leave less room for using optimal viewpoints. Designing an arrangement that acquires optimum perspectives for action recognition while minimizing the influence of viewpoint variety is an unsolved challenge that calls for further inquiry and

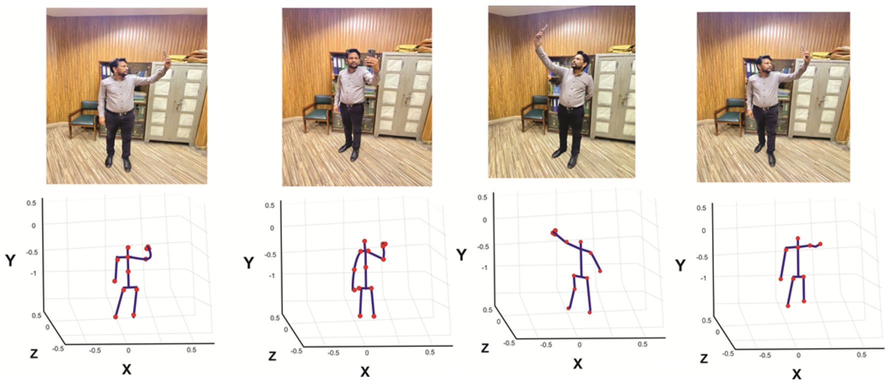


Fig. 1 How the similar posture is shown differently (camera angle and position) makes the skeleton representations look distinctive.

evaluation of existing theories.

In this study, we will attempt to solve the problem of perspective variation to obtain extreme acts for skeleton-based action recognition systems. To eliminate perspective variances, it is better to pre-process the 3D barebones based on human-defined standards. We present a sight adaption technique that sets the observation viewpoint for each sample within the network automatically. It helps the classification unit to "see" the skeleton representation from a different viewpoint, assisting rapid recognition. Note that changing the camera's viewpoint is comparable to transforming the skeleton into a new synchronized structure. We develop an endwise adaptable neural network for viewing, as shown in Figure 2.

For view adaption, its main classification network and sub-network. The observation adaption subnetwork spontaneously governs the related effective perspectives according to input skeletons. The main classification network is then loaded by the new observation views' skeletons, allowing for easier action recognition. The complete network is trained from start to finish, as well as the view alteration subnetwork and the main classification network, to maximize recognition performance. This encourages the view alteration subnetwork to study and govern optimal effective perspectives. View adaptation methods are applied in both the periodic neural network (VA-RNN) and the convolutional neural network (CNN) to demonstrate the efficacy of our suggested view adaptation mechanism (VA-CNN). The final prediction can also be obtained by fusing the classification scores from these two networks, known as the "two-stream" system VA-fusion.

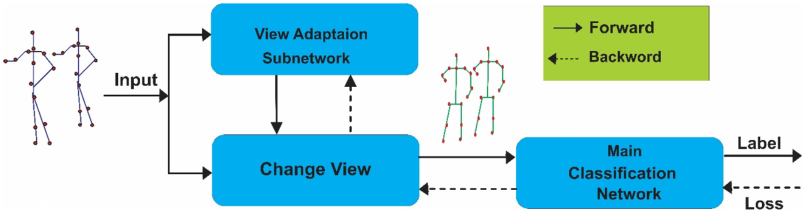


Fig. 2 Endwise sight adaptive neural network flowchart. View adaption network determines virtual observation points. The main classification network classifies skeleton input by transforming it into representations that fit the new points of view.

Our main subscriptions are summed up as follows:

- We present an automatic coordinated view adaption strategy that adaptively repositions observation views to help better recognize actions from skeleton data. This activity reduces pre-processing time and covers several circumstances.
- We design two types of adaptive networks: one is called VA-CNN, and the other is called VA-RNN. For the VA-CNN, we assimilate a CNN-based view adaption component into a CNN organization network for end-to-end learning. During recognition, the view adaptation module chooses each stream's

"optimal" reflection views. In the situation of the VA-RNN, for end-to-end learning, we combine an LSTM classification network with an RNN-based view adaption component.

- We conduct in-depth ablation investigations. Extensive experiments have proved the efficiency of view adaption subnetworks. The impact of various parameters is investigated. Furthermore, we show that the gain is due to our observation alteration component rather than a simple layer increase. We present VA-RNN and VA-CNN modified skeletons to explain why our models work. Discussing failures. We randomly rotate the skeletons during training to make our view adaptation subnetworks more powerful. The findings of our experiments show that data expansion increases the stability of our observation adaption models, which is optimistic.

A successful skeleton-based action recognition structure has been provided by us that spontaneously controls the skeletons to more stable views while keeping an act's continuousness established on the above model improvements and mechanical contributions. We perform view augmentation on the training data to make the view adaption model more robust to changes in the view. Moreover, we do a comprehensive experimental investigation of the network architectures to determine the parameters. Additionally, we run experimentations on five difficult datasets and find that our suggested method constantly attains considerable increases across the board, confirming the efficiency of our suggested models.

2 Related Work

2.1 RGB-based View Invariant Action Recognition

Numerous methods have been planned for view-invariant action recognition based on RGB images, although cameras can apprehend human actions from random angles [19-21]. Several panorama models can be trained using multiple viewpoints [22, 23]. For example, The Oriented Gradients approach trains a Bag of Words based on video input from all viewpoints. However, it is costly to capture videos from numerous viewpoints in practice in another approach; view-invariant representations of features are designed [13, 20]. Descriptions include defining self-similarity [20] or curve-based descriptions [24]. Some original video sequences are lost in the descriptors presented in another domain. A knowledge transfer-based approach might also be considered [25, 26].

2.2 Skeleton-based Action Recognition: Viewpoints

Skeleton-based viewpoint action recognition methods are gaining popularity. Skeleton-based human action recognition methods often outperform RGB-based methods. These compact data are less affected by complex backgrounds and viewpoint changes. Graph Convolutional Networks (GCNs) is the best way to use skeleton data [27]. Xu et al. [28] proposed a two-stream model based on the human skeleton and scene images. Chi et al. [29] introduce a graph convolution method based on attention that can capture human action's intrinsic topology, which changes according to the circumstances. In a recent development by Song et al. [30], a more advanced separable convolutional layer was integrated into more primitive fused Multiple Input Branches (MIB) networks. A strong foundation for skeleton-based action recognition using Graph Convolutional Networks (GCNs) was thus developed. Dynamic GCN automatically learns skeletal topology using Context Encoding Network (CeN). The surrounding joints are considered when studying the link between two joints [31].

2.3 RNN for Action Recognition Using Skeletons

In early works, skeletons are used for action recognition via hand-crafted features [6, 18]. Recurrent Neural Networks have been utilized to notice human actions when raw skeleton inputs are used as inputs in recent works using deep learning. In doing so, the networks learn features and model temporal dynamic behavior. Du et al. [17] Hence, this paper proposes a hierarchy of RNNs that splits the human body into five parts, feeds the parts into different subnetworks, and combines the outputs of each subnetwork. Part-based LSTMs are built according to Amir et al.'s model [32], in which separate cells learn eternal representations of context for a specific part rather than the whole body. Zhu et al. [33] propose systematic detection of discriminative skeleton joints in LSTM networks based on group sparse regularization. Introducing a trusted gate is an effective way to minimize the effects of boisterous joints. Space-temporal attention model [34] incorporates consideration mechanisms in the networks that selectively focus on joints within skeletal systems and pay varying degrees of consideration to productions at different times. Similarly, Liu et al. [35] use global contextual and local information to recognize instructive joints. To distinguish between easy-to-encountered actions at the little stages of pathways and hard-to-encountered actions at the great stages of pathways.

2.4 ConvNet for Action Recognition Using Skeletons

To the importance of convolutional neural networks' extraordinary power for organization, numerous current studies [36, 37] A 2D skeleton arrangement changed to two-dimensional images and then classified with convolutional neural networks. Examples of this include [36, 38], which assign synchronization to the three channels in an image by dividing the border identifiers among the rows (or columns) and joint identifiers between the columns (or rows). Dataset figures [36]or order figures are used to normalize coordinate values to 0-255 [38]. Instead of using absolute values of the joints, In He et al. [37] study used reference joints as a basis for multiple image reconstruction (e.g., right hip, left hip, right shoulder, left shoulder). Some authors [39] generate 2D prognosis drawings based on the curves of joints onto various orthogonal planes. Color and coordinate space of 2D (2D coordinates) represent the 5D space [38].

2.5 Transformers in Computer Vision

Unlike recurrent networks, the Transformer is the best neural model for Natural Language Processing (NLP). There are two major issues that it aims to address: (i) processing of very long sequences, which is hard for both LSTMs and RNNs, and (ii) standard RNN architectures usually process sentences one word at a time, one sentence at a time. This makes it hard to process sentences in parallel. The Transformer has the usual structure of an encoder and a decoder, but it only uses multi-head self-attention. In recent years, transformative self-attention has been used in many common computer vision tasks. Wang et al. [36] came up with a changeable non-local hand based on self-attention. This operator can protectant long-range requirements in both space and time, which makes it easier to classify videos accurately. Dosovitskiy et al. (2020) [40] came up with a Vision Transformer (ViT), which shows how Transformers can be used instead of standard image spinning.

3 View Adaptation Modeling

Even for the same action, the skeleton demonstrations are diverse in different views. The intra-class dissimilarities formed by view differences may be even greater than the inter-class differences. Human action recognition is difficult due to the range of capturing views.

We provide an endwise neural network architecture that spontaneously remarks a skeleton layout from novel computer-generated views before action detection to eliminate the effects of multiple points of view, as illustrated in Figure 2. An organization system and a view adaption sub-network are the two components that make up the system. The view adaption module determines the virtual observation viewpoints, which generates a set of updated factors Tt for each time t in the simulation (or T for a sequence). The main classification network converts the input skeletal representation into classification representations under new perspectives. The complete network is endwise trained to enhance classification performance. The character played by the view adaptation module is the problem of thought perception adaption transformation, formulated in the next subsection.

3.1 Problem Statement

The raw 3D skeletons captured by the camera are representations of the camera coordinate system (global coordinate system), with the camera sensor as the coordinate origin. Our new global coordinate system O is defined using the body-centered in the first frame to be unaffected by an action's initial position and to make our analysis easier. Figure 3 shows the new global coordinate system's skeleton representation as our system's input skeleton Vt.

When shooting for television or film, it is possible to observe activity from many views with the passage of time to better recognize, the scenario and

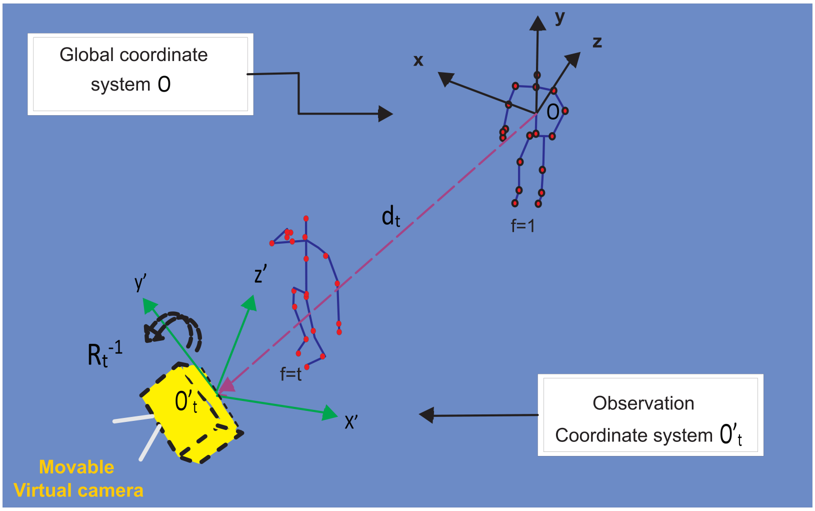


Fig. 3 By estimating that there is a manageable virtual camera, an illustration of the shift in observation viewpoint can be seen. Skeleton sequences are records of the skeletons that have been displayed in frames starting from the first frame, f = 1, to the last frame, f =t, in the worldwide coordinate scheme O. To obtain a new location for the t$_{th}$ frame, and the observation coordinate system must be translated through time by d$_t$ and rotated through time by α_t, β_t, and γ_t, Like the global coordinate system, radians are measured anticlockwise around the X, Y, and Z axes. In the following step, the observation coordinate system O$^{'}$t is utilized to represent the skeleton in three dimensions.

convey a story. By using 3D skeletons taken from a certain vantage point, it is also feasible to maintain an animated virtual camera that can be moved around and monitor the activity from other perspectives, as demonstrated in Figure 3. As shown in Figure 1, When an illustration is observed from the moveable virtual camera viewpoint using the portable virtual camera coordinate system, also known as the observation coordinate system (observation viewpoint), using the skeleton at frame t, the representation below the following coordinate system is transformed into a representation under the following coordinate system: $O^{'}t$.

The j^{th}skeleton joint on the t^{th} framework, assuming a skeleton order S under the global and Z axes. The set of transformation strictures is r coordinate system O, which is denoted as $v'_{t,j} = [x'_{t,j'} y'_{t,j'} z'_{t,j}]^T$ where t $\in(1, \ldots, T)$, j $\in(1, \ldots, J)$, T represents the overall number of frames in order, J represents the whole number of skeleton joints in a frame. The joint's set in the t^{th} frame is denoted as $V_t = \{v_{t,1}, \ldots, v_{t,J}\}$. Assume that the movable virtual camera is located appropriately, for the t^{th} frame, with an associated inspection organize system generated by transformation as $d_t{\in}R_3$, and a turning of $\alpha_t, \beta_t, \gamma_t$, like the global coordinate system, radians are measured anticlockwise around the X, Y, represented by us as $T_t = \{\alpha_t, \beta_t, \gamma_t\}$ For that reason, the representation of jth$skeletonjoint$v'$_{t,j} = [x'_{t,j}, y'_{t,j}, z'_{t,j}]^T = R_t\,(v_{t,j} - d_t)$ of the tth frame under the observation coordinate system O$^{'}$t is.

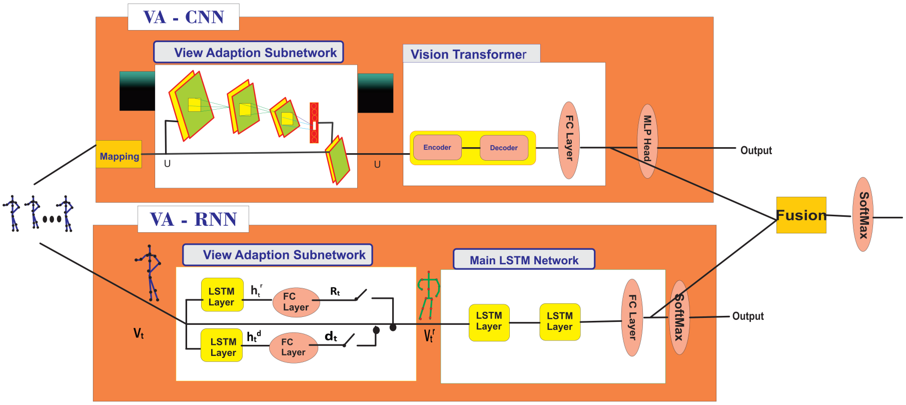


Fig. 4 VA-CNN and VA-RNN have been proposed as two view adaptive neural networks (: a view adaptive CNN (VA-CNN) and a view adaptive RNN with LSTM. Two convolutional networks comprise the VA-CNN: a view adaption subnetwork and a main convolutional network (ConvNet). The view adaption subnetwork is responsible for determining which observation viewpoints are most appropriate for the series. Below the new thought viewpoints, the core ConvNet selects which action class to perform based on the skeleton representations. VA-RNN has two main LSTM networks: a view adaption subnetwork and the main LSTM network. The view adaption subnetwork is responsible for determining the most appropriate opinion viewpoint every time. The action class is determined by the main LSTM network, which uses the skeleton representations underneath new observation views to determine the action class. By combining the classification scores from the two networks the VA-fusion scheme is derived.

$$v'_{t,j} = [x'_{t,j}, y'_{t,j}, z'_{t,j}]^T = R_t\left(v_{t,j} - d_t\right) \tag{1}$$

R$_t$ is symbolized as

$$R_t = R^x_{t,\alpha} R^y_{t,\beta} R^z_{t,\gamma} \tag{2}$$

The coordinate transformation matrixes were represented by $R_t = R^x_{t,\alpha}, R^y_{t,\beta}, R^z_{t,\gamma}$ for rotating the original coordinate system by α_t, β_t and γ_t radians anticlockwise around the X, Y, and Z axes, respectively, which are well-defined as

$$R^x_{t,\alpha} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\alpha_t) & sin(\alpha_t) \\ 0 & -sin(\alpha_t) & cos(\alpha_t) \end{bmatrix} \tag{3}$$

$$R^y_{t,\beta} = \begin{bmatrix} cos(\beta_t) & sin(\beta_t) & 0 \\ sin(\beta_t) & cos(\beta_t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

$$R^z_{t,\gamma} = \begin{bmatrix} cos(\gamma_t) & 0 & -sin(\gamma_t) \\ 0 & 1 & 0 \\ sin(\gamma_t) & 0 & cos(\gamma_t) \end{bmatrix} \tag{5}$$

Furthermore, all skeleton joints have the same transformation parameters as, i.e.,T= $\{\alpha, \beta, \gamma, d\}$,in the tth frame. This is due to changing viewpoints in an inelastic motion. Providing these alteration factors, the skeleton depiction $V_t = \{v_{t,1}, \ldots, v_{t,J}\}$. According to the new findings, coordinates can be found from (1). For different frames, the viewpoints can change throughout time. The main issue is determining the views of the virtual camera that can be moved.

4 View Adaptive Neural Networks

We proposed two different types of adaptable neural networks based on visual perception CNN and RNN, which we refer to as VA-CNN and VA-RNN, correspondingly. As shown in Figure 4. The VA-CNN (as represented at the top) comprises a CNN-based subnetwork for a main convolutional network (ConvNet) and view adaptation. Each network is thoroughly trained by optimizing classification performance from start to finish. The VA-RNN (shown at the bottom) is a view adaptation subnetwork based on RNNs for converting skeletons to new representations under the proper observation views, as well as a primary LSTM network for identifying activities from skeletons that have been transformed. Instead, we may utilize VA-fusion to mix the results from both networks to get a fused forecast.

4.1 View Adaptive Convolution Neural Network (VA-CNN)

A skeleton map makes it easier for ConvNet to model Spatiotemporal dynamics, as shown by the top branch in Figure 4. This is different from VA-CNN, which is shown in the bottom branch. Convolution layers and a fully linked layer are employed to build the view adaption subnetwork. They govern the sequence-level observation perspective, i.e., with altered parameters of α, β, γ, d (as conferred in section 3.1 without subscript). Using the view-adapted skeleton map, a primary ConvNet extracts features for action recognition and investigates spatial and temporal correlations from beginning to end. Image modeling with skeletons.
As in [41], we convert a skeleton arrangement to an image, with columns denoting individual frames and rows showing individual joints. As a result, the 3-dimensional coordinate values for X, Y, and Z are preserved as the three stations of images. We stabilize the pixel standards to be in the range of 0-255 by considering the differences between the 3D skeleton and the image, similar to [41].

$$U_{t,j} = floor\left(255 \times \frac{v_{t,j} - c_{min}}{c_{max} - c_{min}}\right) \tag{6}$$

The variable v$_{t,j}$ is used to denote the 3D coordinates of the jth joint of the tth frame in a skeleton sequence, and the pixel value corresponding to the normalized image map is denoted by the variable u$_{t,j}$. c$_{max}$ and c$_{min}$ are the two maximums and minimums of all the joint coordinates in the training dataset separately, $c_{min} = [c_{min}, c_{min}, c_{min}]^T$ and floor is the maximum integer function.

View Adaptative Subnetwork.

The skeleton representation of the jth joint in the tth frame v$_{t,j}$ is transformed to v$_{t,j}$ according to the transformation formula (1). As a result, the skeleton map's pixel value below the novel reflection viewpoint is estimated as

$$u'_{t,j} = 255 \times \frac{v'_{t,j} - c_{min}}{c_{max} - c_{min}} \tag{7}$$

$$= R_{t,j} u_{t,j} + 255 \times \frac{R_{t,j}(c_{min} - d_{t,j}) - c_{min}}{c_{max} - c_{min}} \tag{8}$$

It can be seen that (8) is formed from (1) and (6) View adaptive network based on CNN is based on CNN learns and determines the observation viewpoint of each of the seven skeleton sequences, then transforms the skeleton map. The view adaption subnetwork is made up of many convolutional covers and a fully connected layer that allows the conversion parameters to retreat, i.e., α, β, γ, d, for R$_{t,j}$, and d$_t$. An altered layer transforms each pixel to a new illustration in the observation viewpoint based on these parameters and (8) in the skeleton map. Therefore, a novel skeleton map is generated to correspond to the new observation viewpoint. We've tried retreating frame-level parameters, as well $T_t = \{\alpha_t, \beta_t, \gamma_t, d_t\}$, For a skeleton map with a width of T-pixels, this translates to 6 × T parameters and retreating order-level parameters $T = \{\alpha, \beta, \gamma, d\}$, which resembles 6 skeleton map parameters. While frame-level parameters appear more elastic and dominant in theory, designing with order-level parameters results in superior performance for ConvNets. Perhaps this is because fewer parameters are at ease to learn.

Vision Transformer (ViT)

As suggested in [42], the established architecture is identical to the original ViT design introduced in [43], except for the MLP head being replaced by a linear classifier. In summary, a ViT model divides the input image into patches. The Transformer encoder is fed a sequence of 1D patch embeddings, and self-attention modules are used to calculate the relation-based weighted sum of the outputs of each hidden layer. Because of this, the Transformers can learn global requirements in the input images as a result of this strategy[43].

The Self-Attention-based Graph Convolution (SAGC) module for spatial modeling and the Multi-Scale Temporal Convolution (MS-TC) module for temporal modeling are the two most important modules in our encoding block. To encode the input and hidden representation of joints, an SA-GC, an MS-TC, a residual connection, and a layer simplification are used[45].

4.2 View Adaptive Recurrent Neural Network (VA-RNN)

The bottom branch of the VA-RNN subnetwork in Figure 4 shows how we employ a view adaption subnetwork to automatically learn and determine observation views, i.e., with transformation factors of $\{\alpha_t, \beta_t, \gamma_t, d_t\}$, (as explained in Section 3.1), and a primary LSTM network for action identification using view-adaptive skeletal data to deduce temporal dynamics and features.

View Adaptation Subnetwork.

The repositioning of the moveable virtual camera (observation coordination system) means that this virtual camera (observation coordination system) can be moved and rotated, which can be considered the customization of the observation viewpoint. It is used to learn the rotation parameters in two branches of LSTM subnetworks, one at a time, one for each frame of the tth frame, and one at a time slot matching the tth frame. $\alpha_t, \beta_t, \gamma_t, d_t$, to acquire the turning matrix Rt and the translation vector d$_t$ $\alpha_t, \beta_t, \gamma_t, d_t$, to obtain the translation vector d$_t$. An LSTM layer and a fully connected (FC) layer are used in conjunction with a branch of the rotation subnetwork to learn rotation parameters. The parameters associated with rotation are derived by utilizing.

$$[\alpha_t, \beta_t, \gamma_t] = \mathbf{W_r h_t^r} + \mathbf{b_r} \tag{9}$$

$h_t^r \in R^{N*1}$ where h_t^r is the unseen output of the LSTM layer, with N being the number of LSTM neurons and ${h^r}_t$ is the vector representing the LSTM layer's hidden output vector $w_r \in R^{3*N}$ and $b_r \in R^{3*1}$ is the FC layer's weight matrix and offset vector, respectively, and the rotation matrices R$_t$ are calculated using the rotational parameters that have been learned (2). The FC layer and the LSTM layer are the two layers that make up the branch of the translation subnetwork used to learn translation parameters. The translation vector d$_t$ is denoted by the symbol.

$$d_t = W_d h_t^d + b_d \tag{10}$$

where $h_t^d \in R^{N*1}$ is the vector representing the LSTM layer's hidden output vector, $w_d \in R^{3*N}$, and $b_d \in R^{3*1}$ represent the FC layer's weight matrix and offset vector, respectively. The skeleton V$_t$ is then represented using the tth frame's observation viewpoint, which is achieved by the use of (1).

Main LSTM Network.

The LSTM network can simulate long-term temporal dynamics and learn feature representations without needing external assistance. As in [34] and [33], Initially, we use two LSTM layers, then one FC layer with a SoftMax classifier, and finally, one FC layer with a SoftMax classifier. This layer has the same number of neurons as action classes in the below layer.

End-to-End Training.

The complete network can be trained from beginning to end. The training loss is cross-entropy loss [34]. Loss gradients go back from each subnetwork to the main LSTM network, not just from the main LSTM network to the view adaptation subnetwork. Let us refer to the loss back propagated to the view adaption subnetwork output as $v'_{t,j} \in R^{1*3}$, where j $\in (1, \ldots, J)$ J is the number of joints in a frame. The loss used to determine d$_t$ translation vector is

$$\epsilon_{dt} = -J\epsilon_{{v'}_{t,j}} R_t, \tag{11}$$

In a similar manner, the loss can be back-propagated to the output of the branch to obtain the rotation parameters. The loss that has been passed back to the output of α_t, for example, is

$$\epsilon_{\alpha t} = \epsilon_{\mathbf{v}'\mathbf{t},\mathbf{j}} \frac{\varrho \mathbf{R_t}}{\varrho \alpha_{\mathbf{t}}} \sum_{\mathbf{j=1}}^{\mathbf{j=J}} (\mathbf{v_{t,j}} - \mathbf{d_t}) \tag{12}$$

With the ability to train from start to finish, the view adaptation model is guided to choose the best observation points to improve the system's ability to recognize things. Instead of relying on human-defined criteria, our suggested system uses an adaptive view adaptation model enhanced for high accuracy recognition to determine the most appropriate observation views based on the content.

4.3 Two Stream Fusion (VA-fusion)

When we combine scores from two different streams, we can use a weighted fusion method to get the final score. This is like the fusion strategy in[46]. Bearing in mind the performance cavity among the two streams, we set the stream for VA-CNN and VA-RNN as 3:2, proven by science. No matter how hard we try, we can't beat the straightforward fusion technique. Among VA-RNN, VA-CNN, and VA-fusion, customers can select the best scheme that fulfills their needs in terms of performance, hardware, and storage space, among other factors.

4.4 Model Implementation and Training

Model Architecture.
To create VA-CNN, we use convolutional neural networks to construct our model. Similar to [43] and [45], we employ EfficientNet for our primary ConvNet, using pre-trained parameters from ViT-b-16 for classification, as described in [43]. Two convolutional layers and one fully connected layer are stacked to create the view adaption subnetwork. The batch normalization layer (momentum is 0.999) and the Relu activation layer are applied after the two convolutional layers. After the second convolutional layer, a max-pooling layer is applied to further reduce the resolution. Finally, an FC (totally connected) or fully connected (FC) layer is applied to predict factors relevant to view transformations. For all of these convolution layers, we set the number of kernels to 128 to achieve the best results. We set the kernel size to 5 and the stride to 2 for each convolutional layer in the model.

$$V^{'} \leftarrow V^{'} \times momentum + v \times (1 - momentum) \tag{13}$$

For VA-RNN, we construct our model utilizing recurrent neural networks using LSTM as the training input. For each LSTM layer, we use 100 neurons to do this. We stack two LSTM layers together to form the primary LSTM network. To learn the transformation parameters for the view adaption subnetworks, we merely use one LSTM layer followed by one fully connected layer, a major reduction in complexity.

View Enhancing via Data-Driven.

All of the data has a restricted number of taking perspectives. This is a typical occurrence, particularly in real-world circumstances. View enrichment is performed at the sequence level to increase the " influence " of our view adaptation model. This is accomplished by spinning the skeleton around the X, Y, and Z axes by a certain number of degrees throughout the training session. Overfitting is predicted to be alleviated, particularly on minor datasets, and the view adaption model will be strengthened due to this.

Model Training.

We train each adaptive neural network with a stream of views from start to finish by minimizing the network's cross-entropy loss. As mentioned in [47], We combine the two streams and use the SoftMax algorithm to calculate the classification probability.

5 Datasets and Experimental Results

Five benchmark datasets, including NTU RGB+D, are used to test our proposed view adaption frameworks, using the NTU RGB+D dataset [32]. In the SYSU dataset, you can see how people interact with objects [48], the Kinetics motion dataset [49], the UCF-101 Motion dataset [50], and the SBU Kinect Contact dataset [51].

Sections 5.1 and 5.2 provide an overview of the datasets and experimental conditions used in this study. In Section 5.3, the proposed view adaption model is evaluated and shown for ablation investigations. In addition, we link our consequences to those of alternative view-invariant methods. The influence of various characteristics is investigated. To better understand the view adaption model, we do an analysis using visualization and explore specific failure instances. A comparison of our results to those of current approaches is shown in Section 5.4, which applies to each of the five datasets examined. The results reveal that our approach steadily outperforms the competition across all datasets. Section 5.5 presents some relative studies of VA-CNN and VA-CNN, divided into two categories.

5.1 Datasets

NTU RGB+D Dataset (NTU) [32]. The dataset of 3960 video trials from the Kinect is the largest data set with RGB+D videos and skeleton data for detecting human actions. It contains 60 different actions plus actions related to regular living, common relations, and physical condition. There are 25 joints in each subject. There is great diversity in sample viewpoints since different angles of cameras, capturing views, and subject orientations are used. In addition, in a cross-sectional evaluation (CS), The 40 subjects are separated into two groups: guidance and analysis are also available, as is a cross-view evaluation (CV), in which cameras two and three samples are used for guidance and camera one for trying. The dataset presents a challenge in act appreciation since there are so many videos and subjects, as well as varying angles of view. **SYSU 3D Human-Object Interaction Dataset (SYSU)** [48]. There are 12 actions in a Kinect dataset taken by 40 individuals. It features a total of 480 scenes. There are 20 joints in each subject. The actions in this dataset share a lot of similarities.

Kinetics-Motion Dataset [49]. A total of 680 video clips (per class) from the most important RGB action recognition dataset was used in the experiment. The dataset contains 30 action classes and 20400 video clips with a duration of 10 seconds per clip. Among those who have contributed to this work are Yan et al. [38], providing joint-based assessed postures for action recognition. The first step was to resize videos to 340,256 pixels at 30 frames per second. There were 30 classes, including skateboarding, tai chi, hopscotch, pull-ups, and capoeira. They also did push-ups and punching bags, belly dancing, country line dancing, surfing the crowd, swimming backstrokes, front raises, crawling babies, and windsurfing. They also did weight lifting, tobogganing, arm wrestling, salsa dancing, and hurling.

UCF-101Motion Dataset [50]. More than 13,000 videos from 101 different types of action are in the folder. They are all at 320 -x 240 resolutions and 25 frames per second. At the same time, the AlphaPose toolbox was used to take about 16 joint actions with RGB videos. On the other hand, Kinetics-Motion has predefined actions like " wounding in the kitchenette " that are more closely linked to specific items and actions in UCF101 than in Kinetics. As many as 3170 videos show 24 different types of exercises that go with the poses. These include a baby crawling on a rope and playing the cello; punching; tai chi; boxing speed bag; pushups; juggling balls; golf swing; clean and jerk; playing the guitar; bowling; ice dancing; juggling balls; bowling; ice skating; and writing on a board.

SUB Kinetic Interaction Dataset [51]. There are 280 skeleton sequences and 6810 frames in this collection. By standard research protocol, we conducted fivefold cross-validation with delivered splits, resulting in eight classes. Two humans were represented by frames in each skeleton, with 15 joints identified for each person. During training, two samples were used for two skeleton sequences, which were then combined. The average predicted score was calculated while the tests were being conducted. Random data collection was utilized to supplement the data collected during the training phase. Five recent crops were selected to calculate the prediction scores, and four bends were more or less for the challenging computation.

6 Experimental Procedure

The batch size for VA-CNN was set at 16. For effective preparation, we set the fully connected layer parameters to zero for the view adaption subnetwork.

All networks are trained with Adam [52], and the early learning rate is set at 0.0001 for all datasets. Skeleton maps have been scaled to 224x224 pixels. To account for the modest sizes of the other datasets, For the NTU dataset, we set the batch size for VA-RNN to 16, and we did the same for the other datasets as well. We initialize the fully connected layer parameters of the view adaptation subnetwork to zeros for this network to promote effective training of the subnetwork. To avoid overfitting Gradient clipping, dropout [45] with a probability of 0.5 is used, which is identical to the method described in [53] and is used to avoid the gradient explosion problem by placing a strict constraint on the gradient's norm (not exceeding 1). All networks are trained with Adam [52] (beta1 and beta2 have values of 0.8 and 0.9, respectively), and the primary learning rate is set to 0.001 for all datasets.

6.1 Affective Ablation Research

Compared to Other Pre-Processing Techniques: Certain methods pre-process the skeletons using human-defined rules to reduce the difficulties created by view fluctuations [17, 34]. We compare these approaches' efficiency with our proposed adaptation model. The NTU RGB+D dataset is now the largest and most demonstrative dataset accessible; we analyze this dataset in depth using recurrent neural networks, and the results are provided in Table 1.

Table 1 Description On the NTU, we compared pre-processing approaches and our view adaption model dataset.

Wo/pre-proc.	Methods	CS	CV
	S-trans+RNN	76.00	82.30
Pre-processing	F-trans+RNN	75.10	80.50
	Raw+RNN	66.30	73.40
	S-trans&S-rota+RNN	76.40	85.40
	S-trans&F-rota (w.r.t shoulder) +RNN	75.80	84.90
	S-trans&S-rota (w.r.t shoulder) +RNN	75.80	85.10
	F-trans&F-rota+RNN	74.10	83.90
	S-trans&F-rota+RNN	75.00	85.10
View-adaptive	**VA-trans +RNN**	**77.4**	**84.4**
	VA-rota+RNN	**87.9**	**94.1**
	VA-RNN	**81.8**	**89.3**

As the RNN architecture changes, VA-RNN is a suggested view adaption technique. This means that the view of the network changes automatically as the network changes. This is our baseline pattern without the view adaption model enabled, which means that the switches s-rota and s-trans are both turned off, resulting in variable Vt being equal to the variable's value. Because the overall coordinate system is moved to the first frame's body center using our view adaption techniques, the input Vt is identical to that used in those methods (see section 3.1). Sequence-level translation, or S-translation, is a sort of pre-processing. Table 1 shows that for the CS and CV settings, the suggested view adaption strategy beats the S-trans+RNN by 5.8% and 7.0 %, respectively, in terms of accuracy. VA-rota+RNN appears to be additional operative than VA-trans+RNN in terms of rotation-only adaptation. In this case, the majority of the actions in this dataset are executed deprived of changing places throughout existence.

If one uses RNN Network skeletons that have been pre-processed based on frequently used human-defined processing criteria, one should be cautious about the performance of the RNN Network. To establish the viewpoints, such pre-processing follow rules that have been defined by humans. The pre-processing-based schemes are denoted by the letters C+RNN, where C denotes the pre-processing strategy, for example, F-trans+RNN. The results of methods that use a range of pre-processing methodologies are shown in the 3rd through 9th rows. F-trans is an abbreviation for frame-level translation, which means that the body center is moved to the origin of the coordinate system for every frame. This rotation is done at the sequence level, and the parameters for the rotation are calculated from the first frame.

The goal is to align the X-axis with the vector running from the "left shoulder" to the "right shoulder," the Y-axis with that of the vector from "spline base" to "spine," and the Z-axis with the new X-Y axis. F-rota, on the other hand, is the frame-wise rotation. During the S/F-rota processing, only the rotation pre-processing required to align the X-axis with the vector from "left shoulder" to "right shoulder" is done at the sequence/frame level (w.r.t shoulder). F-trans&F-rota indicates that both F-trans and F-rota operations are carried out, comparable to the pre-processing carried out in [32]. Using the Raw+RNN scheme in the second row, we can denote a scheme that uses the original skeleton as the input to the RNN Network without performing any pre-processing. It's worth noting that the scale of 3D skeletons is unaffected by the distance between the subject and the camera. As a result, the scaling procedure is not considered in our system. We can draw the following observations and conclusions based on Table 1.

1. Our last strategy outdoes the most frequently used pre-processing solutions by a significant margin. Compared to F-trans&F-rota+RNN [24, 26, 28], our pattern improves inaccuracy by 7.7% and 5.4% for the CS and CV settings, correspondingly. In contrast to Strans&S-rota+RNN, our method improves accuracy by 5.4 percent and reduces inaccuracy by 3.9 percent.

2. Pre-processing at the frame level is inferior to pre-processing at the sequence level because the previous drops more material, such as the gesture diagonally edges, than the latter.

3. Given that it is oblivious to the primary point of action, S-trans+RNN outdoes Raw+RNN, the technique that uses raw skeletons as input by a large margin.

4. As using the CV setting, several human-defined pre-processing options, such as S-transS-rota, S-transFrota, S-transFrota (w.r.t shoulder), and S-transFrota (w.r.t shoulder), produce improved results when compared to when using the CV setting alone. The reason such pre-processing can lessen the diversity of opinions while also resolving the issue of incongruent viewpoints across the training and testing samples.

6.2 Data Augmentation Effects

Through a more inclusive training environment, it is possible to reduce the disparity in opinions between the training and testing environments. It improves the capabilities of the S-trans starting position system and our VA scheme. Data augmentation helps the VA module learn how to modify diverse views by allowing it to "see" more views during training. As illustrated in Table 2, the developed view adaptation model VA and the baseline system S-trans both benefit from data augmentation. The language. Data augmentation is not used in the baseline or proposed view adaption schemes, denoted by S-trans+RNN/CNN and VA-RNN/CNN. The techniques with data augmentation are denoted by the letters Strans+RNN/CNN (aug.) and VA+RNN (aug.)/CNN (aug.).

There is a 3.5 % increase in the performance of CNN-based networks when more data is added to the NTU dataset. This is true for the CS and CV settings of the dataset. Regarding the CV setting, the testing data's perspectives are distinct from the training data's. Therefore, growing the viewpoints in data augmentation can result in a higher gain on the CV setting, allowing some previously unknown testing views to be observed throughout the training process. The view discrepancies between the Kinetic-Motion and UCF101 datasets under the CV setting are substantial, with even a top view. In contrast, the other views are taken by cameras situated roughly horizontally. Data expansion allows the training process to "see" the testing perspectives, resulting in gains of 10.3-12.9%. By enhancing the range of training viewpoints, data augmentation primarily tackles the misalignment between the instruction and testing perspectives.

Furthermore, with the assistance of data increase, VA-CNN (aug.) and VA-RNN (aug.) greatly increased their show compared to VA-CNN and VA-RNN, particularly for the Kinetic-Motion and UCF101 datasets. Data augmentation improves the VA-CNN performance of the Kinetic-Motion and UCF101 datasets by 6.6% and 11.7%, respectively. One of the primary reasons for this is that the VA-CNN and VA-RNN models are incapable of transforming the skeleton order of the testing set into a good learned view when the views of the training and testing sets are significantly out of sync with one another. Although the VA-CNN and VA-RNN models could "see" a large number of views during training, data augmentation made this possible. The models could also translate skeletons from training and testing sets into properly learned views.

Data augmentation is a cost-effective and required strategy to maximize

Table 2 Effectiveness (inaccuracy (%)) of Data-Driven on S-trans and VA schemes.

height	Datasets	NTU		SYSU		SUB	Kinetic-Motion	UCF-101
		CS	CV	Setting-1	Setting-2			
CNN-Based	S-trans+ CNN	86.4	92.4	83.1	81.6	87.9	70.2	74.5
	S-trans+ CNN (aug.)	89.9	93.4	83.2	82.1	87.7	77.3	80.4
	VA-CNN	90.3	95.1	84.3	83.8	90.3	73.8	72.4
	VA-CNN (aug.)	90.1	95.4	88.5	85.9	90.3	80.4	84.1
RNN-Based	S-trans+ RNN	77.4	84.4	77.4	75.3	94.4	86.6	71.02
	S-trans+ RNN (aug.)	79.6	86.1	80.1	79.3	94.9	77.3	73.8
	VA-RNN	81.8	89.3	77.9	77.3	95.4	70.4	76.5
	VA-RNN (aug.)	82.9	92.8	86.4	81.9	98.1	75.1	92.01

efficiency in both the baseline and proposed perspective adaption patterns. Following that, all of our trials are carried out with data augmentation techniques.

6.3 View Adaptation Model Efficiency

Table 3 will relate our suggested VA model to the two prevailing baselines. The baseline techniques are Strans&S-rota+CNN and Strans&S-rota+RNN, including human-defined pre-processing for translation and rotation. For S-trans&S-rota, it's crucial to remember that reversing the skeleton sequence to add extra views is not a good idea in most cases. The goal of the revolution preprocessing is to align the viewpoints. Aside from S-trans+CNN (aug.) and S-trans+RNN (aug.), there are additional sorts of baseline schemes that require translation pre-processing, with no data increase carried out.

View Adaptation versus pre-processing: According to Table 3, the view adaption model consistently outperforms human-defined rotation pre-processing. The pre-processing technique that has been designed by humans is not optimum for recognition performance. Because the human body is non-rigid, the specification of rotation requirements is not always appropriate for the alignment of orientations in space. Our method uses a system to mechanically find the most appropriate views, which has been trained through optimizing ordering correctness.

View adaptation versus data augmentation: From Table 3, VA-CNN (aug.) and VA-RNN (aug.) outperform Strans+CNN (aug.) and S-trans+RNN (aug.) for all datasets. Viewpoint is more of a distraction than a feature that helps you recognize actions. When the training and testing perspectives are identical (for example, after data augmentation), it should be more difficult for a network to deal with different viewpoints because it should be easier to deal with different viewpoints than it should be to deal with only the same viewpoints.VA-RNN is a view adaptation scheme that tries to make the different viewpoints into one consistent viewpoint to make it easier for people to understand. The consistent viewpoint that you've learned is good for learning how to look for specific structures. In summary, our planned VA-CNN (aug.)

and VA-RNN (aug.) schemes steadily outperform two effective baseline plans on all datasets compared to the two powerful baseline schemes.

Table 3 The accuracy of two powerful baseline schemes, S-trans+CNN (aug.) and S-trans+RNN (aug.) with sequence translation preprocessing strategy, S-transS-rota+CNN and S-transS-rota+RNN with sequence translation and rotation preprocessing strategy, and our schemes with view adaptation, was compared (percent). Please note that for both sorts of baseline systems, we highlight the one we believe is the best.

Datasets		NTU		SYSU		SUB	Kinetic-Motion	UCF-101
		CS	CV	Setting-1	Setting-2			
CNN-Based	S-trans&S-rota+CNN	71.1	86.2	76.9	76.2	96.2	77.4	81.1
	S-trans+ CNN (aug.)	89.9	93.4	83.2	82.1	87.7	77.3	80.4
	VA-CNN (aug.)	90.1	95.4	88.5	85.9	90.3	80.4	84.1
RNN-Based	S-trans&S-rota+RNN	82.9	92.8	82.1	80.9	87.4	67.09	77.3
	S-trans+ RNN (aug.)	79.6	86.1	80.1	79.3	94.9	77.3	73.8
	VA-RNN (aug.)	86.9	94.1	86.4	81.9	98.1	75.1	92.01

Influence of Network Parameters: In the recognition network, the VA module is part of it and is responsible for recognition. As a result, the VA-CNN and VA-RNN have more factors than the comparable base networks in their respective domains. One can wonder if the advantages are due to the enlarged number of factors or to the proposed view adaption units, which are both beneficial. There are two techniques to expand a network's model size: (1) Adding more CNN or RNN layers; (2) reducing the number of LSTM neurons or convolutional kernels in RNN-based or CNN-based networks. It should be noted that we are using the EfficientNet as our CNN-based backbone network, using factors pre-trained by ViT. As a result, the number of convolutional kernels has remained the same. For RNN-based networks, we show the results of adjusting the number of LSTM neurons in each RNN layer. The following is a breakdown of the two approaches we'll be taking.

Stacking more layers: The comparisons in Table 4 are between our planned view adaption models and the equivalent primary ordering networks with various layer counts. Each layer of the LSTM architecture has 100 neurons. As the number of LSTM layers in RNN-based networks increases, the performance rises but then declines after two levels. Stacking LSTM layers will not greatly improve performance. Nevertheless, Following the results of a standardized test, our suggested VA-RNN (aug.) scheme outperforms the baseline scheme by using four LSTM layers (2 for the main network, 2 for the VA subnet) with 3,4 or 5 layers by approximately 5.0 percent and 4.9 percent, respectively, in the CS and CV settings, when compared to the baseline scheme with 3,4 or 5 layers. We use EfficientNet of various layers as our support networks for CNN-based networks and discover that a deeper network does not provide noticeable benefits. In both the CS and CV scenarios, our 53-layer system beats the baseline scheme with 152 layers by 3.5 percent and 1.8 percent, respectively, compared to the baseline scheme.

Table 4 The accuracy (percent) of the main ConvNet (S-trans+CNN (aug.)) on the NTU dataset was compared to the accuracy (percent) of the main LSTM network (Strans+RNN (aug.)) on the same dataset using varying numbers of convolutional layers.

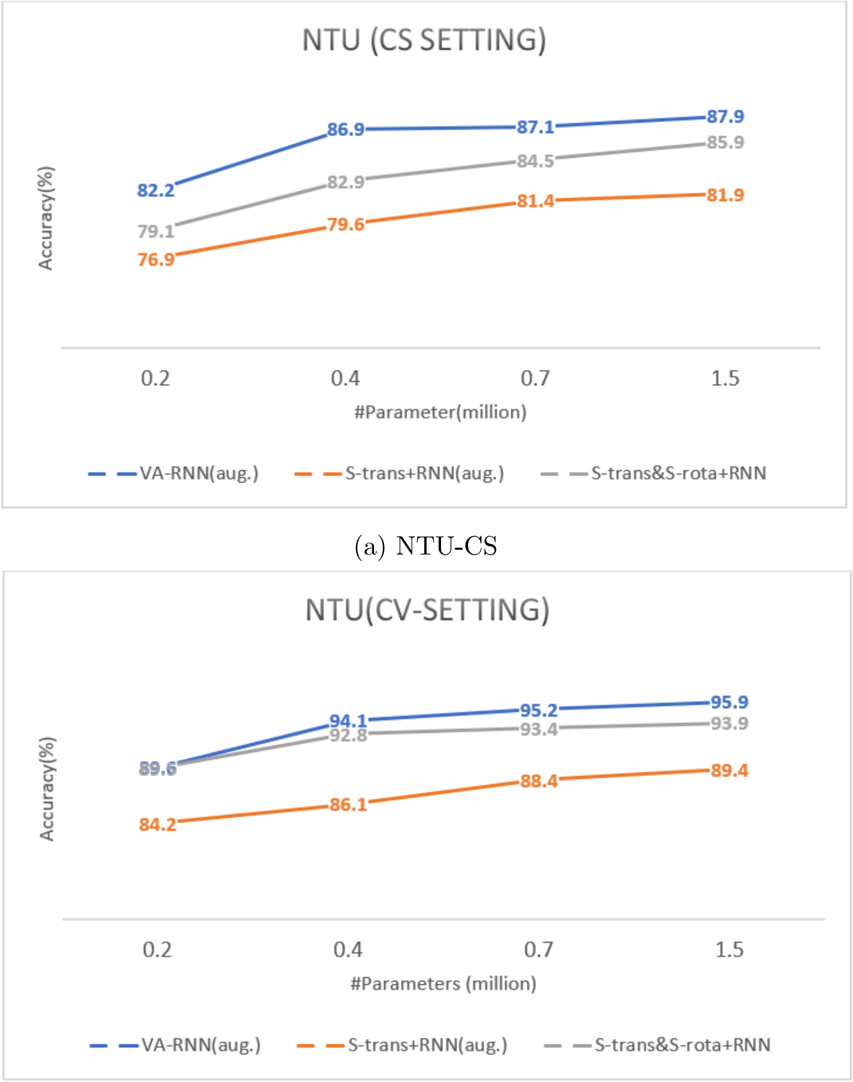
Main Network	Structure	#Param. (M)	CS	CV
S-trans+RNN (aug.)	1 LSTM layer	0.11	83.2	92.1
	2 LSTM layers	0.19	84.3	92.9
	3 LSTM layers	0.27	82.3	91.2
	4 LSTM layers	0.35	82.1	90.6
	5 LSTM layers	0.43	81.9	89.8
VA-RNN (aug.)	2+2 LSTM layers	0.32	86.9	94.7
S-trans+CNN (aug.)	EfficientNet-B5	30	90.7	94.1
	EfficientNet-B6	43	92.4	95.7
	EfficientNet-B7	66	93.2	96.8
VA-CNN (aug.)	EfficientNet-B5+5 linear layers	33.56	96.7	98.6

Increasing the number of LSTM neurons: Comparing our suggested view adaption models to the appropriate primary ordering networks for RNN-based networks with varying numbers of LSTM neurons is shown in Figure 5. The VA model has four LSTM layers compared to the two in the baseline models. Three models were created: VA-RNN (aug.), S-trans+RNN (aug.), and S-trans&S-rota+RNN, each with different numbers of neurons (20, 40, 70, and 150). When we utilize more neurons in a model, the number of factors and the overall show of the model both rise. When using comparable or fewer parameters, VA-RNN (aug.) consistently beats S-trans+RNN (aug.) and S-transS-rota+RNN when using alike or rarer factors for both the CS and CV settings. Observe that the bigger the number of neurons in the LSTM layer, the better the ability to describe the evolution of action dynamics. By default, we utilize 100 neurons for each LSTM layer.

Finally, merely stacking more layers or employing a bigger number of neurons to increase the effectiveness of parameters is not as successful as our planned view adaption module, which is more efficient. Our models outdo the baseline models while having a similar number of parameters.

6.4 Analysis and Visualization of Learned Views

By repositioning the virtual moveable camera, the view adaption subnetworks determine the observation views and then change the input skeleton Vt to the representation Vt below the new viewing platform to optimize the appreciation shown. The illustrations Vt and Vt are shown graphically to help us better comprehend our models. Figure 6 shows the skeletons from various orders, each taken from a unique perspective of (a) a related position or (b) a similar action. A variety of original skeletons representing different points of view are displayed in the second row. Skeletons from our VA-RNN model are presented in the third row after they have been altered. Even when dealing with a wide range of topics and acts, the updated skeletons exhibit substantially more consistent points of view than their original counterparts. As indicated in the fourth row, the skeletons from our VA-CNN model have



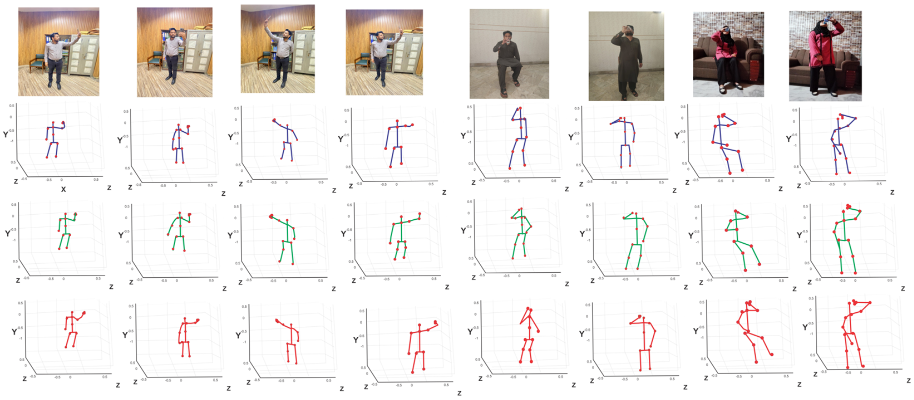
(a) NTU-CS

(b) NTU-CV

Fig. 5 Both baseline and RNN-based view adaptation schemes are shown on the NTU dataset. The curves show how well they work on this dataset. In the recognition network, the VA module is a part of it. The horizontal axis shows how big the model is or how many parameters it has. The vertical axis shows how well the model can be recognized. (%)

been modified. Following an extensive number of observations over various sequences, it has been established that both the VA-RNN and the VA-CCN models are efficient in translating skeletons into significantly more consistent viewpoints. It's worth noting that the most significant factor in our system's success is the consistency of opinions that follow our model. The redesigned

skeletons have also been examined on a wide number of sequences and shown to be able to preserve the flow of an activity. The view adaptation model alters the perspective of a skeleton sequence in response to the contents of the sequence. On the surface, many frames are necessary for the system to fully learn the transformation parameters VA-RNN. Our research shows the following results from our examination of various sequences and transformation factors. First and foremost, as soon as the network receives the first skeleton frame, it begins to modify the skeleton. On the other hand, the learned views of the first few frames are not particularly stunning. As far as the first few frames are concerned, the LSTM network hasn't "seen" nearly enough information to generate a reliable forecast about which views are being presented. It takes anything between 5 and 20 frames to transform the skeleton into something that appears to be very steady.
As illustrated in Figure 7, under the CV condition, the performance of the VA-RNN model outperforms that of the S-trans+RNN model in terms of accuracy (percentage) on the NTU dataset. According to [32], The Id of action is represented by the index of the horizontal axis as [32]. Waving one's hand, for example, depicts the motion represented by the number "23."



(a) (b)

Fig. 6 Edges of (a) the same pose taken since altered perspectives for the similar focus, and (b) the similar action "drinking" taken from altered perspectives for different subjects. Original skeletons in the second row. 3rd row: skeleton representations of our VA-RNN model's observation viewpoints. 4th row: skeleton illustrations of our VA-CNN model's observation viewpoints.

Failure Case Discussion: Those interested in the failure rates of the VA-RNN or VA-CNN models compared to the S-trans+RNN or S-trans+CNN baseline schemes might be interested in learning more. Following extensive research, we discovered that our proposed models are proficient in converting skeletons into stable perspectives, even for samples that had been misclassified before. Using the NTU dataset in the CV condition, Figure 7 shows the histogram of our VA-RNN model's performance improvements over our baseline model S-trans+RNN. To determine the gain values for each action class, subtract the S-trans-RNN accuracy from the VA-RNN accuracy and divide the result by two. We can undoubtedly perceive that our system outdoes the baseline scheme in most classes. We all know that when multiple initialization seeds are used for the same network, performance varies. The VA-CNN method exhibits similar behavior, which we do not illustrate to save space.

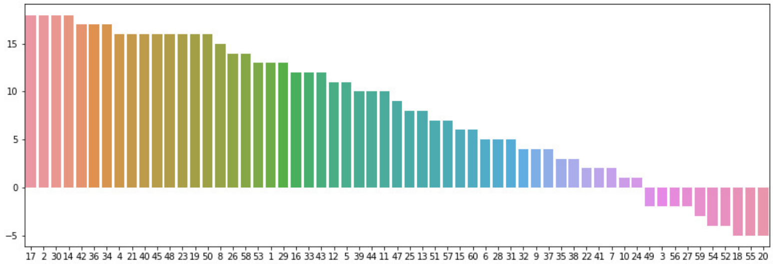


Fig. 7 Shows the VA-RNN model's performance gain in terms of accuracy (percent) for the S-trans+RNN on the NTU dataset for the CV configuration. According to [38], the index of the horizontal axis denotes the Id of action. The action of hand waving, for example, is denoted by the number "23."

6.5 Compared with Other Approaches

In the following segment, we relate our VA-fusion(aug.) stream view adaptation technique to additional state-of-the-art methods on such datasets. Moreover, the performance of VA-RNN (aug.) and VACNN (aug.) is shown.

NTU Dataset: During the performance assessment, we track the ordinary CV and CS conventions proposed by [32]. Our approach is compared with Deep Learning methodologies incorporating RNNs or CNNs and skeleton data [17, 27, 28, 35, 39]. Some traditional approaches use handcrafted elements [54, 55]. The results are displayed in Table 5. This dataset has hundreds of viewpoints, making action recognition difficult. VA-RNN(aug.) and VA-CNN(aug.) outpace standard arrangements and more RNN- and CNN-based systems that apply innovative techniques [17, 27, 28, 35, 39] for instance, attention [34, 39]. In both CS and CV contexts, our planned method's performance of VA-fusion(aug.) outdoes the best state-of-the-art outcomes.

SYSU Dataset: Evaluation of the performance is done under [48]'s standard protocol. Setting 1 requires that half of the subjects be trained, and half of them be tested. As part of Setting 2: A partial of the movies are used for teaching, whereas the second half is utilized for testing. The normal of everything 20-fold cross-validation results for each setting are found in Table 6. With

Table 5 Comparative analysis of proposed and existing models using NTU datasets.

Methods	CS	CV
Skeleton Quads [54]	38.6	41.4
Lie Group [55]	50.1	52.8
Dynamic Skeletons [48]	60.2	65.2
HBRNN-L [56]	59.1	64.0
Part-aware LSTM [32]	62.9	70.3
STA-LSTM [34]	73.4	81.2
GCA-LSTM [35]	74.4	82.8
Clips+CNN+MTLN [37]	79.6	84.8
ESV (Synthesized + Pre-trained) [39]	80.0	87.2
VA-CNN (aug.)	90.1	95.4
VA-RNN (aug.)	82.9	92.8
VA-fusion (aug.)	93.2	97.6

view adaptation models minimizing the view differences, our methodology gets the finest show, which is 14.1% and 6.5% higher compared to [48] for setting-1 and setting-2, correspondingly, and 13.1% greater than [28] for setting-1.

Table 6 Comparative analysis of proposed and existing models using SYSU datasets.

Method	Setting-1	Setting-2
LAFF [57]	54.2	-
Dynamic Skeletons [48]	75.5	76.9
ST-LSTM+True Gate [28]	76.5	-
VA-CNN (aug.)	88.5	85.9
VA-RNN (aug.)	89.1	86.9
VA-fusion (aug.)	89.6	87.2

Kinetic-motion Dataset: To evaluate the performance, we use the standard protocol described by [3]. There are 4 views. The dataset is divided in various ways, yielding 12 partitions. Each division contains three viewpoints: two for training and one for testing. The findings for each partition are shown in Table 7. Because the four perspectives are so dissimilar, it's difficult to tell what's happening when you can't see what's happening. VA-RNN (aug.) and VA-CNN (aug.) meaningfully outstrip zero arrangements using the view adaption model. Although ESV [39] combines 10 separate models, our solo model VA-CNN(aug.) outdoes ESV [39] by 5.5 percent.

Table 7
Comparative analysis of proposed and existing models using kinetic-motion datasets.

Methods	Accuracy (%)
RGB CNN [58]	70.4
Flow CNN [58]	72.8
ST-GCN [59]	72.4
VA-CNN (aug.)	80.4
VA-RNN (aug.)	75.1
VA-fusion(aug.)	83.1

UCF-101 Dataset: In this dataset, there are three views. Two perspectives are often utilized for instruction and the other for testing [60, 61]. Only samples from the first two views are used as training [61], while the third is used for testing. [60] selects every two views as training, resulting in three examples. Comparisons of performance are shown in Table 8. V1 signifies the partition in which training samples are collected from views 2 and 3, and testing samples are taken from views 1. V2 signifies that models from view 2 are being used as trying models. Our VA-fusion(aug.) system obtains the greatest show of 95.3 percent for the V3 setting using view adaption modules.

Table 8
Comparative analysis of proposed and existing models using UCF-101 datasets.

Methods	Accuracy (%)	RGB
LRCN [62]	81.6	True
3D-ConvNet [63]	75.2	True
Two-Stream [43]	91.3	True
DS-LSTM [49]	87.33	True
VA-CNN(aug.)	84.1	True
VA-RNN(aug.)	92.01	True
VA-fusion(aug.)	95.3	True

SUB Dataset: We make use of the ordinary etiquette described by [51], which involves 5 folders and cross-validation. Performance comparisons are shown in Table 9. Our method surpasses previous approaches by a large margin [17, 28, 33, 34] with an accuracy of 99.1 percent. Although this dataset contains no significant view changes, our model tends to detect minor view modifications and turns them into more appropriate views for more efficient action recognition. VA-RNN outperforms VA-CNN in this short dataset (just 282 sequences). Because VA-CNN has a far larger number of parameters than VA-RNN, it's simple for ConvNet to overfit a short training dataset.

Table 9 Comparative analysis of proposed and existing models using SUB datasets.

Methods	Accuracy (%)
Raw skeleton [51]	49.7
Joint feature [51]	80.3
Raw skeleton [56]	79.4
Joint feature [56]	86.9
HBRNN-L [17]	80.4
Co-occurrence RNN [33]	90.4
STA-LSTM [34]	91.5
ST-LSTM + Trust Gate [28]	93.3
GCA-LSTM [35]	94.1
Clips+CNN+MTLN [37]	93.6
VA-CNN(aug.)	90.3
VA-RNN(aug.)	98.1
VA-fusion(aug.)	99.1

The suggested view adaptation module is particularly operational in selecting relevant views, as it was adjusted to optimize recognition performance. The barrier posed by the diversity of views in video recording is overcome by the consistency of viewpoints for diverse actions/subjects, allowing learning action-specific properties to be focused on by the network. Distinct from many other pre-processing techniques, these keep crucial motion information.

6.6 Comparative Analysis of VA-CNN and VA-RNN

As a result of the overview of the view adaptation modules, both VA-CNN and VA-RNN attain perfection in contrast with their standards, as presented in Table 3.

VA-CNN(aug.), In general, is far more dominant than VA-RNN(aug.), as demonstrated in Table 3. The fundamental cause for this is that we convert the complete frame order to an image, allowing a CNN network (such as EfficientNet) to investigate three-dimensional and time-based relationships of the hinges nearby and internationally. The history information is stored in RNN's memory, although restricted.

Table 10 View adaption design effectiveness (inaccuracy (percentage)) on various support CNN networks. S-trans+CNN(aug.) is the baseline system for pre-processing order-level conversion and data amplification.

Networks	Method	#Param.(M)	CS	CV	CS gain	CV gain
CNN-5layers	S-trans+CNN(aug.)	2.06	90.7	94.1	2.1	4.2
	VA-CNN(aug.)	3.56	92.8	98.3		
EfficientNet-B5	S-trans+CNN(aug.)	30	90.7	94.1	2.1	4.2
	VA-CNN(aug.)	30.43	92.8	98.3		
EfficientNet-B6	S-trans+CNN(aug.)	43	92.4	95.7	1.8	1.4
	VA-CNN(aug.)	43.45	94.2	97.1		
EfficientNet-B7	S-trans+CNN(aug.)	66	93.2	96.8	1.2	1.2
	VA-CNN(aug.)	66.42	94.4	97.9		

In comparison to RNN networks, the gain of the view adaption module appears to be less. To evaluate the efficiency of the view adaption module, we run tests on numerous back CNNs with varied model sizes. Table 10 displays the results of CNN-5layers for classification, which contains five convolutional levels and one FC level, as well as our backbone networks EfficientNet-B5, EfficientNet-B6, and EfficientNet-B7.

Two conclusions have been reached. (1) When the CNNs are small, our model produces significant gains. Our view adaption model for the CNN-5layers network obtains improvements of 2.1 percent on the CS and 4.2 percent on CV sets of the NTU dataset, respectively, which are similar to the additions of the RNN-based network (see Table 3). Table 11 contains a summary of all dataset outcomes. When the networks are tiny, we can see that our opinion adaption replicas outperform CNN baselines for all datasets. (2) When the backbone CNNs are huge, our view adaption module sees significant gains. It becomes tougher to achieve the same gain as the model size or complexity grows. Table 10 shows that raising the baseline model size by the same amount results in a lesser increase. In comparison to improving the deepness of the network, a small rise in the perfect dimension of the opinion adaption component boosts performance significantly.

Table 11 Affectivity (incorrectness (%)) of the view adaptation model with small 5layered network CNN- and Efficient-B5, big net as the backbone CNN Networks. the gap between S-trans+CNN (aug.) and VA-CNN (aug.) is represented by Gain.

Network	Method	NTU	SYSU	SYSU	
	CS	CV	Setting-1	Setting-2	Setting-2
EfficientNet-B5					
	S-trans+CNN (aug.)	89.9	93.4	83.2	82.1
	VA-CNN (aug.)	90.1	95.4	88.5	85.9
	Gain	0.2	2.0	5.3	3.8

	Method	SUB	Kinetic-motion	UCF-101
EfficientNet-B5	S-trans+CNN (aug.)	87.7	77.3	80.4
	VA-CNN (aug.)	90.3	80.4	84.1
	Gain	2.6	3.1	3.7

Table 12 compares the number of parameters in our concluding VA-RNN (aug.) and VA-CNN (aug.) modules, evaluating speed and accuracy when the batch size is 1 (number of arrangements in one second) (percent). It's worth noting that deep CNN outperforms RNN with three LSTM layers. The performance of RNNs improves only slightly when more layers are used. We're going to suppose that the arrangement is 300 frames long. (1) VA-RNN (aug.) has the benefit of having a modest model dimension (number of parameters), which is just 2% of VA- CNN's (aug.). (2) VA-CNN(aug.) has a relatively fast recognition speed on well-trimmed sequences, 83.3 orders in one second, which is 10 times faster than VA-RNN (aug.). Because the LSTM configuration is appropriate for the frame-to-frame handling, whereas VA-CNN(aug.) must use a downhill window method to develop flowing data that is untrimmed, VA-RNN(aug.) may be more time effective for the online detection task, based on the sliding window's size VA-CNN(aug.) is around 83.3 structures per second if the window slithers for respectively surround. In contrast, VA-RNN(aug.) is around 7.9300 = 2370 edges in one second if the window glides for every frame. (3) VA-CNN (aug.) has an advanced credit accurateness than VA-RNN because of its combined Spatiotemporal exploration capacity, CNN construction power, and greater model size (aug.). However, due to its short number of parameters, VA-RNN (aug.) performs better for small datasets. Users can pick from VA-RNN (aug.), VA-CNN (aug.), and VA-fusion depending on the needs of actual applications (aug.). TABLE 12: Comparisons of models VA-RNN and VA-CNN.

Table 12 Model comparisons of VA-RNN and VA-CNN

Model	#Param. (M)	Speed (seq./sec.)	Acc. (NTU-CV) (%)	FLOPs
VA-CNN (aug.)	33.56	105.4	95.4	2.7B
VA-RNN (aug.)	0.32	6.1	92.8	2.9B

7 Conclusion

To recognize human action from skeleton data that can handle very long sequences, we present new state-of-the-art end-to-end view adaptive neural networks, VA-RNN and VA-CNN, with the backbone ViT. These two streams are the best for getting a high final score instead of using the standards set by humans to adjust skeletons for action recognition. The proposed networks can adapt the observation perspectives to the most appropriate ones, with the optimization goal of maximizing the recognition performance. We have developed transformer-based view adaptation models based on the recurrent neural network and the convolutional neural network. Both models can transform the skeletons automatically to consistent viewpoints, which reduces the impact of the models' different perspectives and makes training convenient. Experimental results show that the proposed framework consistently improves recognition performance and can handle long sequences on five challenging benchmark datasets. It also gets state-of-the-art results, even though some classes don't perform well because of the model's limitations. Future models with limited datasets will be generalized using the Generative Adversarial Network (GAN).

Ethical Approval: Not applicable as there is no human and or animal data involved.

Competing Interests: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding: Not applicable.

Availability of Data and Materials: The codes and data are available under request from the authors.

Author's Contributions: Methodology and formal analysis done by Faisal Mehmood, Enqing Chen, and Touqeer Abbas; Project administration and Data curation is done by Muhammad Azeem Akbar; and the final revision and English polishing is done by Arif Ali Khan. All authors have read and agreed to the published version of the manuscript.

References

[1] R. Poppe, ”A survey on vision-based human action recognition,” Image and vision computing, vol. 28, no. 6, pp. 976-990, 2010.

[2] D. Weinland, R. Ronfard, and E. Boyer, ”A survey of vision-based methods for action representation, segmentation, and recognition,” Computer vision and image understanding, vol. 115, no. 2, pp. 224-241, 2011.

[3] H. Rahmani and M. Bennamoun, ”Learning action recognition model from depth and skeleton videos,” in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 5832-5841.

[4] L. L. Presti and M. La Cascia, "3D skeleton-based human action classification: A survey," Pattern Recognition, vol. 53, pp. 130-147, 2016.

[5] J. K. Aggarwal and L. Xia, "Human activity recognition from 3d data: A review," Pattern Recognition Letters, vol. 48, pp. 70-80, 2014.

[6] F. Han, B. Reily, W. Hoff, and H. Zhang, "Space-time representation of people based on 3D skeletal data: A review," Computer Vision and Image Understanding, vol. 158, pp. 85- 105, 2017.

[7] F. Mahmood, K. Abbas, A. Raza, M. A. Khan, and K. PW, "Three dimensional agricultural land modeling using unmanned aerial system (UAS)," International Journal of Advanced Computer Science and Applications, vol. 10, no. 1, pp. 443-449, 2019.

[8] Z. Zhang, "Microsoft kinect sensor and its effect," IEEE multimedia, vol. 19, no. 2, pp. 4- 10, 2012.

[9] I. R. https://software.intel.com/en-us/realsense, " " (in), , vol. , no. , , p. , , Art no. , doi: . .

[10] J. Shotton et al., "Real-time human pose recognition in parts from single depth images," in CVPR 2011, 2011: Ieee, pp. 1297-1304.

[11] J. Zhang, W. Li, P. O. Ogunbona, P. Wang, and C. Tang, "RGB-D-based action recognition datasets: A survey," Pattern Recognition, vol. 60, pp. 86-105, 2016.

[12] X. Ji and H. Liu, "Advances in View-Invariant Human Motion Analysis: A Review in IEEE Trans. on Systems," Man, Cybernetics, vol. 40, no. 1, 2010.

[13] F. I. Bashir, A. A. Khokhar, and D. Schonfeld, "View-invariant motion trajectory-based activity classification and recognition," Multimedia Systems, vol. 12, no. 1, pp. 45-54, 2006.

[14] A. Farhadi and M. K. Tabrizi, "Learning to recognize activities from the wrong view point," in European conference on computer vision, 2008: Springer, pp. 154-166.

[15] 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.

[16] A. Razzaq, T. Moughal, M. Zia, S. Qadri, and S. Muhammad, "ROBUST KINEMATIC SKELETON OF HUMAN 3D MODEL IN VIEWING

STRAIGHT LIMBS," Pakistan Journal of Science, vol. 70, no. 4, p. 342, 2018.

[17] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1110-1118.

[18] 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, 2012: IEEE, pp. 20-27.

[19] J.-g. Feng and J. Xiao, "View-invariant human action recognition via robust locally adaptive multi-view learning," Frontiers of Information Technology Electronic Engineering, vol. 16, no. 11, pp. 917-929, 2015.

[20] I. N. Junejo, E. Dexter, I. Laptev, and P. Púrez, "Cross-view action recognition from temporal self-similarities," in European Conference on Computer Vision, 2008: Springer, pp. 293-306.

[21] D. Weinland, M. Özuysal, and P. Fua, "Making action recognition robust to occlusions and viewpoint changes," in European Conference on Computer Vision, 2010: Springer, pp. 635-648.

[22] X. Wu, H. Wang, C. Liu, and Y. Jia, "Cross-view action recognition over heterogeneous feature spaces," in Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 609-616.

[23] A. Iosifidis, A. Tefas, and I. Pitas, "View-invariant action recognition based on artificial neural networks," IEEE transactions on neural networks and learning systems, vol. 23, no. 3, pp. 412-424, 2012.

[24] C. Rao and M. Shah, "View-invariance in action recognition," in Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, 2001, vol. 2: IEEE, pp. II-II.

[25] R. Li and T. Zickler, "Discriminative virtual views for cross-view action recognition," in 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012: IEEE, pp. 2855-2862.

[26] J. Liu, M. Shah, B. Kuipers, and S. Savarese, "Cross-view action recognition via view knowledge transfer," in CVPR 2011, 2011: IEEE, pp. 3209-3216.

[27] Li, R., Wang, H. (2022). Graph convolutional networks and LSTM for first-person multimodal hand action recognition. Machine Vision and Applications, 33(6), 1-16.

[28] Xu, Q., Zheng, W., Song, Y., Zhang, C., Yuan, X., Li, Y. (2021). Scene image and human skeleton-based dual-stream human action recognition. Pattern Recognition Letters, 148, 136-145.

[29] Chi, H. G., Ha, M. H., Chi, S., Lee, S. W., Huang, Q., Ramani, K. (2022). InfoGCN: Representation Learning for Human Skeleton-Based Action Recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 20186-20196).

[30] Song, Y. F., Zhang, Z., Shan, C., Wang, L. (2022). Constructing stronger and faster baselines for skeleton-based action recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence.

[31] Ye, F., Pu, S., Zhong, Q., Li, C., Xie, D., Tang, H. (2020, October). Dynamic gcn: Context-enriched topology learning for skeleton-based action recognition. In Proceedings of the 28th ACM International Conference on Multimedia (pp. 55-63).

[32] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "Ntu rgb+ d: A large scale dataset for 3d human activity analysis," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 1010-1019.

[33] W. Zhu et al., "Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks," in Proceedings of the AAAI conference on artificial intelligence, 2016, vol. 30, no. 1.

[34] 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," in Proceedings of the AAAI conference on artificial intelligence, 2017, vol. 31, no. 1.

[35] J. Liu, G. Wang, P. Hu, L.-Y. Duan, and A. C. Kot, "Global context-aware attention lstm networks for 3d action recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1647-1656.

[36] Wang, X., Girshick, R., Gupta, A., He, K., 2018. Non-local neural networks, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7794–7803. doi:10.1109/CVPR.2018.00813.

[37] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, "A new representation of skeleton sequences for 3d action recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 3288-3297.

[38] B. Li, Y. Dai, X. Cheng, H. Chen, Y. Lin, and M. He, "Skeleton based action recognition using translation-scale invariant image mapping and multi-scale deep CNN," in 2017 IEEE International Conference on Multimedia Expo Workshops (ICMEW), 2017: IEEE, pp. 601-604.

[39] M. Liu, H. Liu, and C. Chen, "Enhanced skeleton visualization for view invariant human action recognition," Pattern Recognition, vol. 68, pp. 346-362, 2017.

[40] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al., 2020. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.

[41] B. Mahasseni and S. Todorovic, "Latent multitask learning for view-invariant action recognition," in Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 3128-3135.

[42] B. Gheflati and H. Rivaz, "Vision Transformer for Classification of Breast Ultrasound Images," arXiv preprint arXiv:2110.14731, 2021.

[43] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. arXiv preprint arXiv:1607.06450, 2016.

[44] 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.

[45] Chi, H. G., Ha, M. H., Chi, S., Lee, S. W., Huang, Q., Ramani, K. (2022). InfoGCN: Representation Learning for Human Skeleton-Based Action Recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 20186-20196).

[46] L. Wang et al., "Temporal segment networks: Towards good practices for deep action recognition," in European conference on computer vision, 2016: Springer, pp. 20-36.

[47] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," The journal of machine learning research, vol. 15, no. 1, pp. 1929-1958, 2014.

[48] J.-F. Hu, W.-S. Zheng, J. Lai, and J. Zhang, "Jointly learning heterogeneous features for RGB-D activity recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 5344-5352.

[49] F. Mehmood, E. Chen, M. A. Akbar, and A. A. Alsanad, "Human Action Recognition of Spatiotemporal Parameters for Skeleton Sequences Using MTLN Feature Learning Framework," Electronics, vol. 10, no. 21, p. 2708, 2021.

[50] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," arXiv preprint arXiv:1212.0402, 2012.

[51] K. Yun, J. Honorio, D. Chattopadhyay, T. L. Berg, and D. Samaras, "Two-person interaction detection using body-pose features and multiple instance learning," in 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2012: IEEE, pp. 28-35.

[52] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.

[53] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," Advances in neural information processing systems, vol. 27, 2014.

[54] G. Evangelidis, G. Singh, and R. Horaud, "Skeletal quads: Human action recognition using joint quadruples," in 2014 22nd International Conference on Pattern Recognition, 2014: IEEE, pp. 4513-4518.

[55] R. Vemulapalli, F. Arrate, and R. Chellappa, "Human action recognition by representing 3d skeletons as points in a lie group," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 588-595.

[56] Y. Ji, G. Ye, and H. Cheng, "Interactive body part contrast mining for human interaction recognition," in 2014 IEEE International Conference on Multimedia and Expo Workshops (ICMEW), 2014: IEEE, pp. 1-6.

[57] J.-F. Hu, W.-S. Zheng, L. Ma, G. Wang, and J. Lai, "Real-time RGB-D activity prediction by soft regression," in European Conference on Computer Vision, 2016: Springer, pp. 280- 296.

[58] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," Advances in neural information processing systems, vol. 25, 2012.

[59] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton- based action recognition," in Thirty-second AAAI conference on artificial intelligence, 2018.

[60] Y. Du, Y. Fu, and L. Wang, "Representation learning of temporal dynamics for skeleton- based action recognition," IEEE Transactions on Image Processing, vol. 25, no. 7, pp. 3010-3022, 2016.

[61] J. Wang, X. Nie, Y. Xia, Y. Wu, and S.-C. Zhu, "Cross-view action modeling, learning and recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 2649-2656.

[62] J. Donahue et al., "Long-term recurrent convolutional networks for visual recognition and description," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 2625-2634.

[63] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in Proceedings of the IEEE international conference on computer vision, 2015, pp. 4489-4497.