Dimensional Expansion and Time-Series Data Augmentation Policy for Skeleton-Based Pose Estimation

SUNG-SOO PARK, HYE-JEONG KWON, JI-WON BAEK, AND KYUNGYONG CHUNG

1Department of Computer Science, Kyonggi University, Suwon-si, Gyeonggi-do 16227, South Korea
2Division of AI Computer Science and Engineering, Kyonggi University, Suwon-si, Gyeonggi-do 16227, South Korea

Corresponding author: Kyungyong Chung (dragonhci@gmail.com)

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2022R1F1A1068828).

ABSTRACT Human pose estimation has long been researched as a significant topic in computer vision. However, studies via deep learning models are insufficient because of the lack of 2D and 3D skeleton data in various domains. An augmentation technique is applied to solve the problem of data scarcity. Data augmentation techniques can improve the performance of an analytical model by increasing the amount of data. However, the model’s performance is degraded if the augmented results differ significantly from the actual distribution. Therefore, it is necessary to implement an optimized augmentation policy for image datasets. This study proposed a dimensional expansion and time-series data augmentation policy for pose estimation based on skeletons. The proposed method improves the model’s performance by using 3D skeleton data. The 3D skeleton data were preprocessed through an affine transformation. The data were augmented for dimensional expansion from 2D to 3D data. In addition, sampling was applied to the data in consideration of time-series features, and thus, the number of frames per unit time was redefined. Subsequently, part of the information was lost by cutout, and thus, the data size, rather than the data shape, was changed. Through the image and video augmentation policies and cutout expansion, search candidates for 3D time-series data augmentation policies were extracted from the number of cases generated through the combination of 16 skeleton augmentation policies, 11 probabilities, and ten intensities. Finally, 20 candidates were extracted, and the five best-performing policies were applied.

INDEX TERMS Data augmentation policy, dimensional expansion, pose estimation, skeleton data.

I. INTRODUCTION

People’s postures vary depending on their individuality. Bad posture causes joint and spinal diseases in the long term. Therefore, it is important to maintain correct posture. When traveling long distances by airplanes, buses, subways, or other means of transportation, people unconsciously adopt unstable postures, resulting in joint and muscle pain. Therefore, it is necessary to identify bad postures and correct them. In computer vision, human skeleton data recorded in a joint unit is used. For two-dimensional (2D) and three-dimensional (3D) pose estimation, the posture information of an image at its photographic time is used [1], [2]. It is possible to predict the future actions of a person from a single image and obtain semantic knowledge about the type of action the person performs. Generally, the research in action recognition in computer vision is based on a heuristic method to determine whether the bone in between joints is placed at a certain angle or if a joint is moved to a specified position. In the expansion of the method and using the skeleton-data-based deep learning model, the technique of recognizing human actions is researched. However, skeleton-data-based deep learning recognition has not been extensively researched owing to data scarcity, and thus its commercialization speed has been slow. Therefore, it is necessary to solve the data scarcity problem based on skeleton data in a general environment [3].
The data labeling process is executed repeatedly to use the large amount of data obtained as input data for deep learning. Because such data is sufficiently large for deep learning researchers, a method using the collective intelligence of numerous people has also been developed. The amount of skeleton data shared on the Internet is small; therefore, data is used only for development research or in a local environment. In addition, it is difficult to visualize the data, and thus, it is difficult to perform labeling with the use of collective intelligence. Consequently, the data sets for research are open source. However, the volume of data in these datasets for deep learning is not sufficiently large, and many cases of inaccurate performance measurement exist. Therefore, a deep learning model based on skeleton data uses the Kinetics-Skeleton and NTU RGB+D 120 datasets as the scale of performance comparison [4], [5].

- The Kinetics-Skeleton dataset is a set of converted 2D skeleton data for pose estimation based on Openpose from the video-based dataset Kinetics 400. This dataset consists of massive amounts of data and various types of actions. However, with the low accuracy of pose estimation, a significant amount of information loss occurs. Therefore, it is used for sketchy performance measurements. In addition, in a deep learning model that is learned with 2D data, if the 3D data recorded with a depth sensor is received as input data in a deep learning model, it must be recognized correctly [6], [7]. Therefore, by using the 3D dataset of NTU RGB+D 120, it is necessary to develop a high-performance deep learning model.

- The NTU RGB+D 120 is designed to add 120 action data to the NTU RGB+D dataset, which is composed of 3D skeleton data for 60 actions. The performance of action recognition based on NTU RGB+D 120 data [8] has been researched using various deep learning models. Accordingly, the accuracy rate improved from 60% to 80% within approximately one year. Nevertheless, it is necessary to develop a variety of data preprocessing techniques.

Image data scaling up or down, transformation, and other techniques are used in the conventional vision field. Therefore, many methods have been studied. In the case of image data, performance can be improved through preprocessing and augmentation techniques. It means that preprocessing and augmentation techniques are expected to improve performance for skeleton data. Given that image data is 2D, applying such a technique to 3D skeleton data is difficult. Skeleton data can be used to predict a posture, motion, or action during a certain time, and it is necessary to consider time-series characteristics with temporal information. Therefore, it is necessary to research preprocessing and augmentation techniques considering both dimensional and time-series characteristics [8], [9].

This study proposed a dimensional expansion and time-series data augmentation policy for pose estimation based on skeletons. The proposed technique normalizes skeleton data through 3D affine transformation and then redefines the augmentation policy for fast auto augmentation. Thus, it improves the performance of an action-recognition model for pose estimation. In terms of conventional policies, it redefines shear, translate, rotate, flip, and resize to fit the affine transformation-based augmentation technique in three dimensions. Regarding the augmentation policies applied to a video recognition model, up-sampling and down-sampling are both redefined to fit the skeleton data. In the image data augmentation policies, a cutout with high performance is redefined to fit the skeleton data. An optimal policy was found among these 16 augmentation policy candidates. Accordingly, by normalizing the distribution imbalance caused by data scarcity, a deep learning model with high performance can be learned.

This paper is organized as follows: chapter 2 describes the action recognition model and image data augmentation technique with the use of skeleton data, and the search for an optimal augmentation technique through reinforcement learning; chapter 3 describes the proposed dimension expansion and time-series data augmentation policy for pose estimation based on skeletons; chapter 4 describes the search for an optimal data augmentation technique for pose estimation and its performance evaluation; and chapter 5 concludes the paper.

II. RELATED WORK
A. BEHAVIOR RECOGNITION MODEL USING SKELETON DATA

Because the NTU RGB+D dataset of actions and NTU RGB+D 120 dataset of 120 action data among 3D skeleton data were open-source, action recognition models for performance improvement were studied [8]. A conventional skeleton data-based action recognition model converts joint coordinates into colors using a deep learning model for image recognition. In addition, the converted images with the coordinates of the frames and joints were used as input data. Owing to the loss of joint coordinates and time-series information, accurate analysis is difficult. Although overfitting learning is applied to a small amount of data, a low accuracy (less than 50%) is obtained. With the development of natural language processing and improvements in RNN models, the features of time-series information can be extracted. It is used as a typical model for skeleton data-based action recognition [10]. A part-aware LSTM was proposed using the NTU RGB+D dataset. The performance rate was 70%. To extract 3D data features, models based on diverse techniques were used. The GCN-based ST-GCN showed high performance; therefore, the graph convolution network became a representative recognition model. In a graph, a joint is defined as a node, and a bone-linking joint appears similar to an edge. Therefore, it is impossible to extract features accurately [11]. Joint features are extracted in ST-GCN to recognize the actions. Furthermore, 2s-AGCN, which uses an ensemble of the results from the models learned for joints and bones, was
proposed, and showed performance improvement. In addition, a preprocessing technique for 3D skeleton data was proposed. However, performances were not compared depending on whether the data were preprocessed. Thus, the influence of preprocessing on performance was not observed [12]. Therefore, the two-stream method is used to separate the joints and bones of the skeleton data, extract features with AGCN, and combine the results to recognize the behavior. In addition, MS-G3D, which designs a model through diversely modified GCN, further improved the performance. After separating the joints and bones of the skeleton data, it extracts their features in MS-G3D, and combines the extraction results to recognize actions. The data preprocessing method was the same as that of 2s-AGCN, and no data augmentation technique has been proposed [13]. Through an improved data preprocessing technique and optimal augmentation policy, performance improvement can be expected. Fig. 1 shows the architecture of the skeleton-data-based action recognition model.

C. EXPLORING THE OPTIMAL AUGMENTATION TECHNIQUE USING REINFORCEMENT LEARNING

Optimal augmentation policies and preprocessing techniques for data have long been studied and are still challenging research problems. If transformed data are unrealistic or severely distorted, they deteriorate the model’s performance. Therefore, it is necessary to consider the data characteristics and collection circumstances before selecting an appropriate augmentation technique for data. However, this process requires considerable time and cost. In addition, it is difficult to reflect on all cases; thus, it cannot be called optimal. Therefore, it is challenging to apply frequent heuristic searches to a commercial deep-learning model. AutoAugment has been proposed to solve this problem. AutoAugment is an algorithm for automatically finding an augmentation policy suitable for a dataset by using reinforcement learning. SOTA performance was achieved with an optimal augmentation policy of a dataset [16]. The search space used for the reinforcement learning of AutoAugment is defined as follows:

- To search for an augmentation technique suitable for data in diverse ways, one policy is composed of five sub-policies.
- One sub-policy has two operations. An operation is an augmentation technique. Sixteen techniques are determined using 11 probabilities and 10 intensities.

The number of augmentation techniques for one sub-policy is the value obtained by multiplying $16 \times 11 \times 10$ twice. As for the policy composed of five sub-policies, there are approximately $2.9 \times 10^{32}$ candidates. In terms of performance, it was impossible to check all these cases. Therefore, an algorithm that reduces the search depth and width is required.

To train a model, AutoAugment transforms the training data using randomly selected multiple augmentation techniques. Repeatedly, it selects a case to improve the model’s performance and randomly chooses a technique. Therefore,
it searches for the policy, including an optimal augmentation technique, and applies it to the model for performance improvement. If the training data distribution is insufficient for validation data distribution, data augmentation supplements the shortage to improve performance. Therefore, the performance was improved when AutoAugment was applied to a small data set rather than a large dataset [17]. Compared to the performance improvement, the search cost is high. Fast AutoAugment was used to solve this problem. To reduce the search cost of AutoAugment, Fast AutoAugment changes augmentation target data from training data to validation data to search for a policy. It trains a model with training data, puts the validation data reflecting augmentation policies into the model, chooses a policy with the highest performance, applies the augmentation policy to the training data, and enables the model to relearn to improve performance. By decreasing the model learning count, it is possible to reduce search costs. Therefore, for the optimal augmentation policy search of the ImageNet dataset, the algorithm shows performance improvement, similar to AutoAugment, and reduces search costs [18].

### III. DIMENSIONAL EXPANSION AND TIME-SERIES DATA AUGMENTATION POLICY FOR SKELETON-BASED POSE ESTIMATION

The dimension expansion and time-series data augmentation policy for pose estimation based on skeletons proposed in this study consists of five steps, namely, data normalization-based preprocessing; data augmentation for dimension expansion; data sampling using time-series features; data loss-based cutout; and deep learning-based data augmentation. It searches for an optimal data augmentation technique for pose estimation.

In data normalization-based preprocessing, the 3D skeleton dataset is preprocessed through 3D affine transformation and 3D time-series data normalization. In the data augmentation step for dimension expansion, data is augmented based on the fast autoaugmentation policy and image data augmentation. In the data-sampling step using time-series features, the time unit expressing a value is newly defined. In the data loss-based cutout step, a rectangular area is set randomly, and part of the pixel information is lost. Because the data has a different optimal size, part of the information is lost, and the loss is used for analysis. In the deep learning-based data augmentation step, the probability distribution of the pose data is estimated using a variational autoencoder (VAE), and the skeleton of the frame that appears after the initial frame is generated.

From skeleton data augmentation policy candidates, an optimal data augmentation technique is searched for using heuristics and reinforcement learning. Fig. 2 shows the process of the proposed dimension expansion and the time-series data augmentation process for pose estimation based on skeletons.

#### A. PREPROCESSING USING DATA NORMALIZATION

The training data used in this study is a set of 3D skeleton data for RGB+D human action recognition. It consists of the data presented as 25 3D joint coordinates in space. It is the NTU RGB+D, having 120 human actions and containing approximately 100,000 3D skeleton data obtained with a depth sensor, which is Kinect v2. It is relatively large; therefore, it is useful in a deep learning model. Compared to other types of data, 3D skeleton data is relatively small [8]. Their data distribution is imbalanced owing to direct photography. A data preprocessing method can be applied to all data, including learning and validation data. For all data, a preprocessing method without data information loss was applied for the generalized distribution of a recognition model.

Normalization is possible through coordinate movement, rotation, or other kinds of transformation. To achieve this, an affine transformation is applied. An affine transformation is used to preserve the coordinate information in a space and transform a shape using a one-to-one correspondence function. It transforms data without any loss if there is no limited value range. If the range of data expression is limited, data loss can occur when a value exceeds the range that results from transformation. An affine transformation is used for 2D image processing, and methods such as translation, scale, shear, and rotation are used. The 3D affine transformation is implemented through the expansion of the 2D affine transformation. Equation (1) shows the 2D and
3D affine transformations.

\[
2D = \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & t_1 \\ a_{21} & a_{22} & t_2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}
\]

\[
3D = \begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & t_1 \\ a_{21} & a_{22} & a_{23} & t_2 \\ a_{31} & a_{32} & a_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}
\]

In (1), \( x \), \( y \), and \( z \) represent the original coordinates; \( a \) and \( t \) are the variables; \( x' \), \( y' \), and \( z' \) represent the coordinates after transformation.

The 3D affine transformation performs that in space.

The 3D affine transformation transforms data to a matrix using Numpy supporting matrix operations and defines the transformation matrix for fast operations. Skeleton data are time-series data of 3D coordinates generated over a continuous period. Generally, transformation operations need to be applied at every point in time. If Numpy broadcasting is used, it is possible to operate the transformation of skeleton data quickly through 3D affine transformation [19]. The skeleton data recorded by a depth sensor are 3D time-series data of 25 joint coordinates in space. The starting point of the skeleton data is the center of the depth sensor, rather than the basis of a particular joint. Even with the same information, the direction an object considers and the center position are different depending on the position between an object and a depth sensor. Therefore, if information is expressed consistently through data normalization, the generalization of a recognition model using the data can occur. Accordingly, the following data normalization method was proposed and applied to the 2s-AGCN:

- The spine joint (#2) of the first frame and first person was translated as (0, 0, 0).
- The straight line connecting the hip joint (#1) and spine joint (#2) of the first frame and the first person was rotated parallel to the z-axis.
- The straight line connecting the left shoulder (#5) and right shoulder (#9) of the first frame and the first person was rotated parallel to the x-axis.

First, the affine transformation was used for the translation operation to move to particular coordinates. Using the inverse vector generated with the differences between the coordinates of the first frame before and after translation, an affine transformation is designed and applied at all times. Equation (2) shows the affine transformation matrix for the translation of the spine joint (#2) to the zero point of coordinates. In the equation, \( j_2 \) represents the coordinates of the spine joint (#2).

A multiplication operation is applied to the translation matrix \( T \) for all frames and joints, as follows:

\[
j_2 = \begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix} T = \begin{pmatrix} 1 & 0 & 0 & -x_2 \\ 0 & 1 & 0 & -y_2 \\ 0 & 0 & 1 & -z_2 \end{pmatrix}
\]

Next, with the use of affine transformation, a rotation operation was executed to obtain two straight lines running parallel to each other. The cross product of two straight lines running parallel to each other in the first frame was set as the rotation axis. Using the included angle of the two straight lines, the affine transformation was designed and applied to all times. Equations (3)-(7) show the affine transformation matrix that is rotated to form the straight-line connecting hip joint (#1) and spine joint (#2) in parallel with the z-axis or matched with the axis.

\[
v_1 = \overrightarrow{j_1j_2} = (x_2 - x_1, y_2 - y_1, z_2 - z_1)
\]

\[
v_2 = k = (0, 0, 1)
\]

Equation (3) shows the vectors of the two straight lines running parallel to each other. In the equation, \( v_1 \) represents the straight line connecting the hip joint (#1) and spine joint (#2), and \( v_2 \) represents the z-axis.

\[
\cos \theta = \frac{v_1 \cdot v_2}{\|v_1\| \|v_2\|} = \frac{x_1 x_2 + y_1 y_2 + z_1 z_2}{\sqrt{x_1^2 + y_1^2 + z_1^2} \sqrt{x_2^2 + y_2^2 + z_2^2}}
\]

\[
\sin \theta = \sqrt{1 - \cos^2 \theta}
\]

In (4) and (5), the dot product of two straight lines is used to represent the trigonometric functions \( \cos \theta \) and \( \sin \theta \) of the included angle.

\[
u = (u_x, u_y, u_z) = \frac{v_1 \times v_2}{\|v_1 \times v_2\|}
\]

Equation (6) shows the unit vector of the cross product of two straight lines and the axis of rotation \((C=I - \cos \theta)\). Equation 7, as shown at the bottom of the page.

Equation 7 shows the affine transformation matrix with respect to the unit vector of the cross product, which is rotated by the included angle \( \theta \) as the rotation axis. With the use of affine transformation, a rotation operation is executed to obtain a straight line, connecting the left shoulder (#5) and right shoulder (#9), that runs parallel to the x-axis. Equation (8) shows the straight line connecting the
left shoulder (#5) and right shoulder (#9) and the vector of the a-axis.

\[
v_3 = j\rightarrow (x_9 - x_5, y_9 - y_5, z_9 - z_5)\\
v_4 = i = (1, 0, 0)
\]  

Equation (8) shows the vectors of the two straight lines to be parallel to each other. \(v_3\) represents the straight line connecting the left shoulder (#5) and right shoulder (#9), and \(v_4\) represents the x-axis. In (4), (5), (6), and (7), as shown at the top of the previous page, by substituting \(v_1\) and \(v_2\) for \(v_3\) and \(v_4\), respectively, it is possible to calculate the axis of rotation and the included angle, and to create an affine transformation matrix.

**B. DATA AUGMENTATION USING DIMENSIONAL SCALING**

The fast autoaugmentation policies used for image augmentation include affine transformation-based techniques, image color-based techniques, and image damage techniques. Among them, affine transformation-based techniques are shear, translation, and rotation, all of which can be expanded to 3D affine transformation. Because these methods are applicable to all times, they are suitable for time-series data [20]. Fig. 3 shows the affine transformation-based dimension expansion involved in the fast autoaugmentation policies.

- **Shear** is a transformation method for designating new coordinates according to the degree of shearing. Two-dimensional ShearX designates a new x coordinate according to the shearing degree \(\alpha\) and y coordinate; ShearY specifies a new y coordinate according to the shearing degree \(\beta\) and x coordinates. The shearing value of the 3D ShearX is an x value. A new x value is designated according to the degree of shearing of the y and z values. The degree of freedom is two, according to the degree of shearing. However, to use a fast autoaugmentation policy, one value is used according to the degree of shearing. Therefore, 3D ShearX designates a new x coordinate according to the shearing degree \(\alpha\), y, and z coordinates. When a new x coordinate has the largest change, it is \(\sqrt{2}\alpha\) at \(y = z\). In the Fast AutoAugment, the degree of shearing ranges from –0.3 to 0.3. Any change outside this range can also be outside the range of the data set distribution. Therefore, the available range of the shearing degree is from \(-0.3 \leq \sqrt{2}\alpha \leq 0.3\). Therefore, the range of the shearing degree \(\alpha\) is set to \(-0.21 \leq \alpha \leq 0.21\), and the ranges of \(\beta\) and \(y\) for ShearY and ShearZ are the same as those of \(\alpha\).

- **Translate** is a transformation method for designating new coordinates according to the degree of translation. Two-dimensional TranslateX designates a new x coordinate according to the translation degree \(\alpha\), and similarly TranslateY specifies a new y coordinate. Three-dimensional TranslateX, TranslateY, and TranslateZ have equal degrees of freedom, which is one, and can use a Fast AutoAugment policy. Three-dimensional TranslateX, TranslateY, and TranslateZ designate new coordinates according to the translation degrees \(\alpha\), \(\beta\), and \(\gamma\). The degree of translation defined in Fast AutoAugment policies ranges from –150 to 150 and is equal to 45% of the image size. If Translate is executed, the information from a range of image sizes is lost. The 3D skeleton data had no range of expression. Therefore, any translation to a large value causes no information loss. Being extremely far from the zero point of coordinates lowers the accuracy of the data analysis. To prevent data from being transformed significantly, it is necessary to compare the absolute value of the largest value with the absolute value of the smallest value in all frames and to set a range to half the smaller value.

- **Rotate** is a transformation method for designating new coordinates through rotation. RotateX, RotateY, and RotateZ, the expanded 3D axes from 2D ones, have an equal degree of freedom, which is one, and can use an AutoAugment model. Therefore, RotateX, RotateY, and RotateZ designate new coordinates, according to

---

**FIGURE 3.** Dimensional expansion using affine transformation included in Fast AutoAugment policy.
the rotation angle $\theta$. In Fast AutoAugment, the rotation angle ranges from $-30^\circ$ to $30^\circ$, which is the same as in the three dimensions.

Image augmentation techniques not involved in Fast AutoAugment policies vary, including affine transformation, perspective transform, contrast, Gaussian noise, color change, and cutout. Among them, affine transformation-based methods, such as Flip and Resize, are easily applicable to 3D skeleton data. These methods are expandable to 3D affine transformations. Because they are applicable to all times, they are suitable for time-series data. Fig. 4 shows the affine transformations that are not involved in Fast AutoAugment policies.

- **Flip** is a transformation method for designating new coordinates using fling coordinates on the zero point. By multiplying the x coordinate by $-1$, it is possible to specify a new coordinate. Its degree of freedom is zero, and the application of a policy becomes a variable. This method is meaningful in that it supplements the area that rotates fail to include in insufficient data. In the three dimensions, by setting the flip reference coordinates to $x$, $y$, and $z$, it is possible to establish a policy. Accordingly, FlipX, FlipY, and FlipZ designate a new coordinate by multiplying the $x$, $y$, and $z$ coordinates by $-1$.

- **Resize** is a transformation method for designating new coordinates according to a multiplying factor from the zero point of the coordinates. In a 2D image, it is used as a type of scale. This policy sets the center of an image as the zero point and multiplies the $x$ and $y$ coordinates using a multiplying factor. It is expandable to the three dimensions by applying the multiplying factor operation to the $x$-, $y$-, and $z$-axes. Resize designates the new $x$ and $y$ coordinates according to the multiplying factor $\alpha$. Although the policy was expanded to three dimensions, its degree of freedom was one. Accordingly, 3D Resize designates new $x$, $y$, and $z$ coordinates according to the multiplying factor $\alpha$. In a 2D image, the value of Resize ranges from 0.5% to 1.5%, or from 50% to 150%. In the case of a 100% excess multiplying factor, the image size is out of its range, and thus information is lost. The 3D skeleton data has no information loss even if the range of the multiplying factor is infinite.

Being far from the zero point of coordinates reduces the accuracy of the data analysis. Therefore, it is necessary to check a range of data after normalization, set up a multiplying factor appropriately, and set its range to 0.5–1.5.

### C. DATA SAMPLING USING TIME SERIES FEATURES

The posture changes gradually over time, incorporating time-series features. Image data augmentation techniques include affine transformation, Gaussian noise, color change, and sampling. Among them, sampling employs time-series features and defines the unit of time that expresses a value. Therefore, it is possible to extend to skeleton data with time-series features. With a sampling rate, the sampling policy redefines the number of frames per unit time [21], [22]. For example, we assume that we have 1,800 frames of a video, which is played for 60 s and is down-sampled to 900 frames through redefinition. In this case, the video is played completely within 30 min at the same playing speed. This appears to be a fast-playing speed effect. If 1,800 frames are up-sampled to 3,600 frames through redefinition, it appears as a slow playing speed effect. The sampling was redesigned into a linear graph. This means that a linear graph including all the existing data is generated and that new values are set up to divide it by a constant distance for new elements. Fig. 5 shows the skeleton data sampling algorithm.

![FIGURE 5. Algorithm of skeleton data sampling.](image)

```
FIGURE 4. Affine transformations not involved in Fast AutoAugment policies.

```

```
```

```c
Input: skeleton, ratio // Skeleton Data, Sampling Ratio
Output: Sampled Skeleton Data

new_len = int(len(skeleton)*ratio) //set new frames length
Array new_frames[new_len]

for i = 0 to new_len
    new_frames[i] = skeleton[int(i/new_len*len(skeleton))]
end // rebuild with new frame number
return new_frames
```

In Fig. 5, the frame length and sampling rate of the original data are operated, the new frame length is defined, and the
data are recorded in line with the rate. Subsequently, the original frames were connected in a line. A new frame is designed based on the line, and the original frames are up-sampled or down-sampled. In this course, information from the original frames is lost partially or duplicated and is then involved in a new part of the data distribution. Excessive sampling causes information damage or severe duplication, which lowers the accuracy of recognition. Therefore, it is necessary to set up an appropriate range of 0.5 to 1.5, the same as in Resize.

D. CUTOUT BASED ON DATA LOSS

Some image augmentation techniques use partial information loss, such as Gaussian noise, cutout, dropout, and salt and pepper. Among these methods, Gaussian noise, dropout, and salt and pepper lose pixel information in a random position of an image as well as set the density and size of a loss position as variables. The cutout sets up the rectangular area at a random position and loses the pixel information. It sets the number of regions and sizes as the variables. Cutout showed the best performance when variables were searched experimentally and was used as a Fast AutoAugment policy candidate. If the method is expanded and applied as a skeleton data augmentation technique, high-performance improvements can be expected. Cutout determines a rectangle area of an image and fills it up with zero. It appears similar to dropout in terms of filling up with zero. Cutout, however, applies such a method to the input data. The performance of Cutout is influenced more by the size of the loss area than by its shape. Because data have a different optimal size, it is necessary to search for the size experimentally. The method of searching for and applying an optimal variable results in the highest performance improvement [23].

To apply Cutout to the skeleton data, the method of filling up the joints for pose estimation in the input with zero is used. The performance of Cutout is influenced by the size of the area filled up with zero. Therefore, performance can vary depending on the number of joints that turn zero. If the number of joints turning zero is extremely large, this information loss negatively influences the performance. Therefore, in Fast AutoAugment, the range of Cutout policy is set to zero to 60, equal to zero to 1/5 of the image size. Because the skeleton data consists of 25 joints, its range is set to zero to five, which is equal to zero to 1/5. Fig. 6 shows the skeleton cutout algorithm.

In Fig. 6, the coordinates of the randomly selected joint are changed to zero.

E. DEEP LEARNING-BASED DATA AUGMENTATION

Deep learning-based data augmentation methods for training neural networks with datasets and generating new data have been actively researched. A deep learning augmentation technique can obtain the potential expressions to re-generate data by allowing a model to learn data features automatically.

Based on the learned potential expressions, realistic data can be generated and data can be collected with high generality [24]. For deep learning-based data augmentation, generative models such as AutoEncoder and generative adversarial network (GAN) are used. These models learn good expressions of data through the structure of encoder and decoder and generate results that are close to the actual data distribution. Compared to GAN, AutoEncoder has a stable learning performance, although it generates data similar to learning data. In summary, the generated data were not new. By contrast, GAN generates more definite data, and the data is similar to real data.

To regenerate skeleton data through deep learning, this study employed the PoseVAE. If several initial frames of a human pose are provided, PoseVAE estimates the probability distribution of future poses. Therefore, it is possible to regenerate the skeleton data in line with the given input frame label. Video action prediction research using conventional PoseVAE adds Pose GAN to obtain a clear pixel video as an outcome. Accordingly, for the augmentation of skeleton data, PoseVAE is applied to obtain stable and new data in line with a given label [25], [26], [27]. Fast AutoAugment experimentally searches for an optimal augmentation policy based on a dataset and a deep learning recognition model. The number of cases where candidates for Fast AutoAugment are used for one policy is $(16 \times 11 \times 10)^2 \approx 29 \times 10^{22}$. Therefore, a total inspection is impossible. Consequently, we minimize the search time by reducing the search depth and width through a searching algorithm using RNN-based reinforcement learning. Candidates are applied to the validation data rather than learning data, and a search is conducted after the performance improvement is checked. Thus, it is possible to search for a policy that improves performance. However, different results were obtained for each search time. If five policies are applied to a model after being searched for, it is possible to expect a similar level of performance improvement.
IV. RESULT AND PERFORMANCE EVALUATION

A. EXPLORING THE OPTIMAL DATA AUGMENTATION TECHNIQUE FOR POSE ESTIMATION

Seventeen skeleton augmentation policies have been proposed with the expansion of the image augmentation policy, video augmentation policy, cutout, and deep learning augmentation policy. To select an optimal augmentation technique, an experimental search and a fast autoaugmentation technique are both applied. In the search method, 16 policies equal to Fast AutoAugment policies used for image data augmentation policy search are determined, based on 11 probabilities and ten intensities. Table 1 shows the augmentation policy candidates for pose estimation based on the skeletons.

| Operation Name | Range of Magnitudes |
|----------------|---------------------|
| Shear(X,Y,Z)   | [-0.21, 0.21]       |
| Translate(X,Y,Z)| [-1/2 * min(MAX, |min(MIN)), 1/2 * min(MAX, [MIN])] |
| Rotate(X,Y,Z)  | [-30, 30]           |
| Flip(X,Y,Z)    |                     |
| Random Resize  | [0.5, 1.5]          |
| Sample (Up, Down)| [0.5, 1.5]        |
| Cutout         | [1, 5]              |

In Table 1, a range of application intensities for each policy candidate is defined and applied. If an optimal route is searched for using 16 policy candidates, it is not fair to conclude that the augmentation method improves the performance the most. However, if limited policies are used, it is possible to find and apply an optimal method with the highest performance improvement.

The performance improvement in augmentation policies depends on the datasets and deep learning recognition models. It is possible to conduct an experimental search to select an augmentation policy with the highest performance improvement. Because 16 techniques, 11 probabilities, and ten intensities are variables, the search count is 16 × 11 × 10 = 1,760. The search cost for the total inspection is extremely high. The number of cases can be lowered by reducing the number of candidates for some variables. Even if an experiment is conducted after probabilities and intensities are designated as random numbers, it is possible to determine whether an augmentation technique improves performance. The cutout policy improves the performance relatively differently depending on the intensity. Therefore, as for the policy, all intensities are searched. If learning is more than a certain level, it is impossible to find the top performance, although it is possible to compare the performance level approximately. Therefore, the epoch of learning is downsized to 1/2 for testing and comparison, and thereby a policy with the highest performance improvement is found. Five candidates with high performance were selected for the policy. A performance improvement can be expected by applying it to a skeleton-based action recognition model.

B. PERFORMANCE EVALUATION

The test and implementation environment of the 3D time-series data preprocessing for skeleton-based action recognition proposed in this study employed two HPC GPU servers with 20TFLOPS. In addition, Python libraries NVIDIA Apex for MS-G3D and PyYAML were used. The test dataset used in this study was NTU RGB+D 120 [8]. Based on NTU RGB+D, approximately 60 types of actions performed with Kinect v1 have 60 types of actions performed with Kinect v2. In summary, it consisted of 114,480 videos. Data information includes RGB, depth, 3DJoints, and IR recorded by Kinect. Among these, 3D joints were used. Cross-subject was applied as a performance measurement method. In addition, the joint-only model of MS-G3D was used, and all parameters except for Epoch retained their existing settings [28], [29]. Table 2 shows the dataset of NTU RGB+D 120.

| Dataset     | Classes | Subjects | Views | Video | Year |
|-------------|---------|----------|-------|-------|------|
| NTU RGB+D 120 | 120     | 106      | 155   | 114,480 | 2019 |

For performance evaluation, in the condition of learning with 60 epochs, the MS-G3D model using preprocessing is compared with the model that does not use preprocessing. In addition, the accuracy was analyzed according to the changes in the X, Y, and Z axes. After random probabilities and intensities were applied to all the suggested data augmentation policies and learning with 30 epochs was processed, the proposed model was compared with a conventional model. In terms of accuracy, the top five policies are mixed for a new policy, and then learning with 60 epochs is processed. In this state, the proposed model was compared with a conventional model. Additionally, in the condition of learning with 60 epochs, the Fast AutoAugment-based model is compared with a conventional model. Finally, the classification performance was evaluated through a confusion matrix.

In terms of performance evaluation, first, the influence of preprocessing of the MS-G3D joint-only model on the performance was evaluated. Table 3 shows the performance evaluation results based on normalization. The value of the learning epochs was 60.

| Model Type         | Top-1 Acc. | Top-5 Acc. |
|--------------------|------------|------------|
| MS-G3D [4, 26]     | 82.38%     | 95.76%     |
| MS-G3D without    |            |            |
| Normalization [ours]| 83.65%     | 96.75%     |

As shown in Table 3, the model without preprocessing shows better performance than the conventional model using
preprocessing [30], [31]. This means that the preprocessing method of the model is designed to imbalance the data distribution. Therefore, it is necessary to develop an improved preprocessing method that fits the data and recognition models.

Second, accuracy was evaluated when shear, translation, rotation, and flip were applied to each of the \(X\), \(Y\), and \(Z\)-axes. Table 4 shows the application results of shear, translation, rotation, and flipping to each of the axes.

As shown in Table 4, when shear, translate, rotate, and flip are applied to each of the axes, the accuracy is approximately 50%. This means that simple changes in the axes fail to exhibit good performance. Therefore, it is necessary to develop an effective data-augmentation policy.

Third, the influence of the proposed 3D data augmentation policy on the performance of the MS-G3D model was evaluated. Table 5 shows the performance evaluation results according to the 3D data augmentation policies. The value of the learning epochs was 30.

As shown in Table 5, ShearY, TranslateZ, Resize, DownSample, and Cutout result in the top-5 performance. This means that performance is improved more when augmentation policies are applied than when they are not applied, and that an augmentation technique balances the data distribution. By contrast, FlipY lowered performance when it was applied, compared to when it was not applied. This means that the policy imbalances the data distribution. The top five augmentation policies with performance improvement were applied for performance evaluation. Table 6 shows the performance evaluation results according to the top five augmentation policies and according to the optimal augmentation policy found with Fast AutoAugment [27].

The value of the learning epochs was 60. As shown in Table 6, the performance is improved more when augmentation policies and a deep learning-based augmentation technique are applied than when they are not applied. Performance can be improved solely through an input data augmentation technique, with no model design changes or parameter adjustments [32]. Although the PoseVAE-based augmentation policy improved performance more than the conventional model, its performance was lower than that of the top five augmentation policies. The top five augmentation policies are transformations that preserve the geometric nature of the data. By contrast, PoseVAE, as a model to predict a future pose, generates data whose geometric nature is changed. If such an augmentation policy is applied, unexpected data can be included in the model training. Therefore, the top five augmentation policies improve the performance more than a deep learning-based augmentation policy. Nevertheless, these top five augmentation techniques may not be optimal; therefore, an experimental search was conducted for an optimal augmentation policy. The model in which an optimal augmentation policy was found and applied with Fast AutoAugment showed better performance than the MS-G3D model.

The method based on the top five augmentation policies improved the performance the most. This means that the policy found with Fast AutoAugment consists of optimal augmentation techniques. Fig. 7 shows the classification results obtained using the confusion matrix. The proposed shows (a) the classification result for MS-G3D + Top5Augment, and (b) the classification result for MS-G3D + Fast AutoAugment as a confusion matrix.

Fig. 7 shows the results of classifying 114,480 images into 120 classes. As a result of the MS-G3D + Top5Augment evaluation, 111,812 results correspond to TP and TN.
Accordingly, the accuracy was approximately 97.67% and the precision was approximately 98.8%. By contrast, MS-G3D + fast AutoAugment evaluation results showed 112,110 results corresponding to TP and TN. Accordingly, the accuracy was approximately 97.9% and the precision was approximately 98.63%. Because the proposed method is suitable for 3D skeleton data processing, its performance can be improved.

### V. CONCLUSION

This study proposed a dimensional expansion and time-series data augmentation policy for pose estimation based on skeletons. This is a method to improve the performance of the behavior recognition model for pose estimation. The proposed method proceeds with the data preprocessing, data augmentation, and sampling steps. In the data preprocessing stage, the affine transformation was used to prevent the loss of information owing to coordinate movement and rotation. In the data augmentation stage, affine transformation and deep learning were used. It can be extended in three dimensions; thus, it can be applied equally at all times. In the sampling step, the unit of time for expressing values was newly defined using the time series feature, and the new values were effectively set by dividing them at regular intervals by the number of new elements. In addition, by using the cutout technique to randomly set a region, information was lost, which affected the performance of the variable search. Accordingly, a total of 16 augmentation techniques were selected so that they could be searched with Fast AutoAugment, and performance evaluation was performed according to 3D skeleton data preprocessing and augmentation techniques. First, in the performance evaluation according to preprocessing, performance improved more when preprocessing was not applied than when it was applied. It was found that preprocessing rendered the data distribution imbalanced. Second, when transformations were made simply according to the axes, the performance was low. Third, in the performance evaluation according to augmentation policies, some augmentation policies lowered performance, whereas ShearY, TranslateZ, Resize, DownSample, and Cutout (4) produced the top five performance results. When these policies were implemented, the performance was high. Finally, Fast AutoAugment was applied to find an optimal augmentation policy, and performance was evaluated when the policy was applied. Performance improved more when the policy found with Fast AutoAugment was applied than when the policies presented in the third evaluation were applied. The proposed 3D time-series data augmentation technique for skeleton-based behavior recognition has proven that the optimal data augmentation policy according to the data set and model is different. Therefore, if the deep learning model is trained through the most appropriate policy to improve the model’s performance, accurate and effective pose estimation is possible.

In future studies, we plan to study performance improvements in different datasets and different recognition models compared to the augmentation policies that can be applied to the proposed 16 additional policies, and how data augmentation policies are generalized in the skeleton-based behavior recognition model.

### REFERENCES

[1] T. L. Munea, Y. Z. Jembre, H. T. Weldegebriel, L. Chen, C. Huang, and C. Yang, “The progress of human pose estimation: A survey and taxonomy of models applied in 2D human pose estimation,” IEEE Access, vol. 8, pp. 133330–133348, 2020.

[2] Y. Chen, Y. Tian, and M. He, “Monocular human pose estimation: A survey of deep learning-based methods,” Comput. Vis. Image Understand., vol. 192, pp. 102897–102919, Mar. 2020.

[3] X. Zhang, C. Xu, and D. Tao, “Context aware graph convolution for skeleton-based action recognition,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 14333–14342.

[4] S. Liu, H. Zhang, Z. Chen, Z. Wang, and W. Ouyang, “Disentangling and unifying graph convolutions for skeleton-based action recognition,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 143–152.

[5] P. Verma, A. Sah, and S. R. Srivastava, “Deep learning-based multi-modal approach using RGB and skeleton sequences for human activity recognition,” Multimedia Syst., vol. 26, no. 6, pp. 671–685, Jul. 2020.

[6] S. Yan, Y. Xiong, and D. Lin, “Spatial temporal graph convolutional networks for skeleton-based action recognition,” in Proc. AAAI Conf. Artif. Intell., 2018, pp. 1–10.

[7] Z. Cao, G. Hidalgo, T. Simon, S. E. Wei, and Y. Sheikh, “OpenPose: Realtime multi-person 2D pose estimation using part affinity fields,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 43, no. 1, pp. 172–186, Jan. 2019.

[8] 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. 2019.

[9] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, “NTU RGB+D: A large scale dataset for 3D human activity analysis,” in Proc. IEEE Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 1010–1019.

[10] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Comput., vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[11] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in Proc. Int. Conf. Learn. Represent. (ICLR), 2017, pp. 1–14.

[12] L. Shi, Y. Zhang, J. Cheng, and H. Lu, “Two-stream adaptive graph convolutional networks for skeleton-based action recognition,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 12026–12035.

[13] G. Zhu, L. Zhang, H. Li, P. Shen, S. A. Shih, and M. Bennamoun, “Topology-learnable graph convolution for skeleton-based action recognition,” Pattern Recognit. Lett., vol. 135, pp. 286–292, Jul. 2020.

[14] J. Zhang, D. Yu, J. H. Liew, X. Nie, and J. Feng, “Body meshes as points,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 546–556.

[15] G. Roger and C. Schmid, “MoCap-guided data augmentation for 3D pose estimation in the wild,” 2016, arXiv:1607.02046.
E. Lashgari, D. Liang, and U. Maoz, “Data augmentation for deep-learning: Learning augmentation strategies from data,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 113–123.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Jun. 2009, pp. 248–255.

S. Lim, I. Kim, T. Kim, C. Kim, and S. Kim, “Fast autoaugment,” in Proc. Adv. Neural Inf. Process. Syst., vol. 32, 2019, pp. 6665–6675.

C. R. Harris, K. J. Millman, S. J. Van Der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, and R. Kern, “Array programming with numpy,” Nature, vol. 585, no. 7825, pp. 357–362, Sep. 2020.

Z.-G. Du, J.-S. Pan, S.-C. Chu, H.-J. Luo, and P. Hu, “Quasi-affine transformation evolutionary algorithm with communication schemes for application of RSSI in wireless sensor networks,” IEEE Access, vol. 8, pp. 8583–8594, 2020.

J. E. Szymanski, Basic Mathematics for Electronic Engineers: Models and Applications, London, U.K.: Taylor & Francis, 1989.

A. V. Oppenheim, Discrete-time Signal Processing, Chennai, India: Pearson, 1999.

S.-X. Chen, S. Sahmani, and B. Safaei, “Size-dependent nonlinear bending behavior of porous FGM quasi-3D microplates with a central cutout based on nonlocal strain gradient isometric finite element modelling,” Eng. Comput., vol. 37, no. 2, pp. 1657–1678, Feb. 2021.

E. Lashgari, D. Liang, and U. Maoz, “Data augmentation for deep-learning-based electroencephalography,” J. Neurosci. Methods, vol. 346, Dec. 2020, Art. no. 108885.

H. Xu, J. Li, H. Yuan, Q. Liu, S. Fan, T. Li, and X. Sun, “Human activity recognition based on gramian angular field and deep convolutional neural network,” IEEE Access, vol. 8, pp. 199393–199405, 2020.

H. Duan, Y. Zhao, K. Chen, D. Lin, and B. Dai, “Revisiting skeleton-based action recognition,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2022, pp. 2969–2978.

S. S. Park, “3D time-series data augmentation technique for skeleton based action recognition,” M.S. thesis, Dept. Comput. Sci., Kyonggi Univ., Suwon-Si, South Korea, 2020.

D.-H. Shin, K. Chung, and R. C. Park, “Prediction of traffic congestion based on LSTM through correction of missing temporal and spatial data,” IEEE Access, vol. 8, pp. 150784–150796, 2020.

D.-H. Shin, R. C. Park, and K. Chung, “Decision boundary-based anomaly detection model using improved AmoGAN from ECG data,” IEEE Access, vol. 8, pp. 108664–108674, 2020.

H. Jung and K. Chung, “Social mining-based clustering process for big-data integration,” J. Ambient Intell. Humanized Comput., vol. 12, no. 1, pp. 589–600, Jan. 2021.

H. Yoo and K. Chung, “Deep learning-based evolutionary recommendation model for heterogeneous big data integration,” KSI Trans. Internet Inf. Syst., vol. 14, no. 9, pp. 3730–3744, Sep. 2020.

H. Yoo, R. C. Park, and K. Chung, “IoT-based health big-data processing technologies: A survey,” KSI Trans. Internet Inf. Syst., vol. 15, no. 3, pp. 974–992, 2021.

SUNG-SOO PARK received the B.S. and M.S. degrees from the Division of Computer Science and Engineering, Kyonggi University, Suwon, South Korea, in 2019 and 2021, respectively. Since 2019, he has been a Founding Member of Gepetto Robotics Company Ltd., South Korea. He has worked as a Researcher at the Data Mining Laboratory, Kyonggi University. His research interests include data mining, deep learning, computer vision, motion detection, and motion estimation.

JI-WON BAEK received the B.S. degree from the School of Computer Information Engineering, Sangji University, South Korea, in 2017, and the M.S. degree from the School of Department of Computer Science, Kyonggi University, South Korea, in 2020, where she is currently pursuing the Ph.D. degree with the Department of Computer Science. She worked with the Data Management Department, Infiniq Company Ltd. She has been a Researcher at the Data Mining Laboratory, Kyonggi University. Her research interests include data mining, data management, knowledge systems, automotive testing, deep learning, medical data mining, healthcare, and recommendations.

HYE-JEONG KWON received the B.S. degree from the Division of Computer Science and Engineering, Kyonggi University, Suwon, South Korea, in 2021, where she is currently pursuing the master’s degree with the Department of Computer Science. She has worked as a Researcher at the Data Mining Laboratory, Kyonggi University. Her research interests include data mining, artificial intelligence, mobile healthcare, biomedical and health informatics, information systems, deep learning, machine learning, emerging health risk mining, and medical data analysis.

KYUNGYONG CHUNG received the B.S., M.S., and Ph.D. degrees from the Department of Computer Information Engineering, Inha University, South Korea, in 2000, 2002, and 2005, respectively. He has worked with the Software Technology Leading Department, Korea IT Industry Promotion Agency (KIPA). From 2006 to 2016, he was a Professor at the School of Computer Information Engineering, Sangji University, South Korea. Since 2017, he has been a Professor with the Division of AI Computer Science and Engineering, Kyonggi University, South Korea. Clarivate Analytics named him a Highly Cited Researcher, in 2017. His research interests include data mining, artificial intelligence, healthcare, knowledge systems, HCI, and recommendation systems.