Hindawi
Mobile Information Systems
Volume 2023, Article ID 9813246, 1 page
https://doi.org/10.1155/2023/9813246

Retraction

Retracted: 3D Human Pose Estimation Based on Transformer Algorithm

Mobile Information Systems

Received 25 July 2023; Accepted 25 July 2023; Published 26 July 2023

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope

(2) Discrepancies in the description of the research reported

(3) Discrepancies between the availability of data and the research described

(4) Inappropriate citations

(5) Incoherent, meaningless and/or irrelevant content included in the article

(6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] G. Chen, "3D Human Pose Estimation Based on Transformer Algorithm," *Mobile Information Systems*, vol. 2022, Article ID 6858822, 9 pages, 2022.

Hindawi
Mobile Information Systems
Volume 2022, Article ID 6858822, 9 pages
https://doi.org/10.1155/2022/6858822

Research Article

3D Human Pose Estimation Based on Transformer Algorithm

Guowei Chen

New Media Institute, Communication University of China, Chaoyang, Beijing, 100024, China

Correspondence should be addressed to Guowei Chen; 201772442@yangtzeu.edu.cn

Received 23 June 2022; Revised 6 August 2022; Accepted 8 August 2022; Published 29 August 2022

Academic Editor: Santosh Tirunagari

Human pose estimation (HPE) is a fundamental problem in computer vision, and it is also the basis of applied research in many fields, which can be used for virtual fitting, fashion analysis, behavior analysis, human-computer interaction, and auxiliary pedestrian detection. The purpose of HPE is to use image processing and machine learning methods to find out the positions and types of joints of people in pictures. There are two main difficulties in HPE. First, the complex human images make the model need to learn a highly nonlinear mapping relationship, and the learning of this mapping relationship is extremely difficult. Second, the highly nonlinear mapping relationship needs to be learned by using a model with high complexity, and a model with high complexity requires a lot of computational overhead. In this context, this paper studies the 3D HPE based on the transformer. We introduce the research status of HPE at home and abroad and provide a theoretical basis for designing the transformer 3D HPE model in this paper. We introduce the technical principle and optimization scheme of CNN and transformer and propose a 3D HPE model based on transformer. We used two datasets, COCO and the MPII datasets, and performed a number of experiments to find the best parameters for model development and then assess the model's performance. The experimental findings suggest that the strategy described in this study outperforms all other methods on both datasets. The average precision (AP) of our model reaches up to 79% on COCO dataset but a PCKh-0.5 score of 81.5% on the MPII dataset.

1. Introduction

When photos of human bodies are analyzed using HPE, computer vision-related technologies are used to extract the body's important features and link them together. Image sensors capture human body images, which are subsequently analyzed by computer vision algorithms to extract important points and the relationships among those points. Finally, analysis extracts the human body's key points and relationships among them. HPE technology has also made significant strides and has been better developed and implemented in recent years with the advancement of software and hardware [1]. Including algorithm improvement and optimization, the new algorithm can analyze the relevant structure of the human body more comprehensively and intelligently and use less resource occupancy to obtain more accurate key point positioning. The improved image quality of the image sensor and the more detailed images can make the human torso and limbs clearer, so as to achieve better analysis results. The improvement of the processor, faster clock frequency, and better performance can improve the processing speed of the algorithm, shorten the execution time of the algorithm, and perform more complex algorithm analysis in the same time [2]. For human pose analysis, traditional artificial methods first need to perform preprocessing methods such as illumination normalization, histogram equalization, and grayscale correction on the image to obtain relatively clear and stable images, then use HOG, SIFT, or morphological processing to obtain human-related features, then normalize these features by visual word bag and other methods, and finally use the processed features to use the HPE algorithm to determine whether there is a human body and the location of key points of the human body, so as to achieve HPE [3]. The traditional manual feature extraction steps are cumbersome and lack high-level semantic information, which makes the HPE under the traditional method limited by the scene, resulting in low accuracy and generalization. It is even more difficult for the occlusion of human key points and the recognition of complex poses. Compared with traditional methods for

HPE, deep learning methods represented by DNNs have been favored by algorithm researchers in recent years. Compared with manual feature extraction, DNN methods have better robustness [4]. Unlike statistical learning methods, machine learning methods require algorithm researchers to have prior knowledge of the corresponding domain and then model the domain according to the prior knowledge and rules. Feature engineering plays an important role in machine learning methods. The construction of simple and effective features is an important basis for judging whether the model is good or bad. At the same time, the feature dimension will also greatly affect the performance of the model. Too few features will make the model unable to fit. If the target problem is met, too many features will make the model overfit. The advantages of automatic feature extraction by deep learning can well deal with feature extraction problems. By designing a model structure based on convolution and nonlinear operations, a large number of effective features can be automatically extracted. At the same time, features are continuously combined and abstracted in the model. It makes the features have more high-level semantic information and global features [5]. The model updates and corrects the parameters through the gradient-based backpropagation algorithm and finally converges to the local optimum point to complete the parameter learning [6]. At the same time, the overparameterized model has an implicit regularization effect, which can alleviate the overfitting problem caused by too many parameters [7]. In short, with the excellent characteristics of DNN algorithms, one of the most popular algorithms in business and academics is deep learning, which has many applications in computer vision, natural language processing, and voice recognition. HPE belongs to the field of computer vision with great research value and challenging direction and has a wide range of applications in military, security, industry, and entertainment, mainly in intelligent video surveillance, patient rehabilitation systems, human-computer interaction, human body animation capture, and virtual reality and more. The HPE algorithm can realize automatic human behavior analysis and action recognition. Compared with traditional manual analysis algorithms, deep learning algorithms greatly improve efficiency and liberate productivity, thereby enabling the automation of the above industries and other related scenarios, reducing the consumption of human resources, and enabling more new scenarios. [8].

The main work of this paper is to study 3D HPE based on transformer, develop a HPE software library, provide convenience for researchers of HPE algorithms and application software developers, and simplify the research and development process of algorithm researchers and related application developers. Reducing the development difficulty of related practitioners can also make more college students and other entry-level developers pay attention to the field of HPE and jointly promote the development and application of this field. We study the 3D HPE based on the transformer and provide a theoretical basis for designing the transformer 3D HPE model in this paper. We introduce the technical principle and optimization scheme of CNN and transformer and propose a 3D HPE model. We take images of human pose and use various data enhancement techniques such as rotation, scaling, and saturation adjustments and use this data to train the HPE model. We used two datasets, COCO and the MPII datasets, and performed a number of experiments to find the best parameters for model development and then assess the model's performance. The experimental results prove the efficiency of the proposed approach.

2. Related Work

Computer vision has a long history of using 3D models to identify objects. 3D models may be used to identify a restricted number of categories, such as vehicles and motorbikes. Design unique characteristics that match the synthetic 3D model with genuine photographs in [9–12]. Some academics have started to use neural networks for 3D object identification because of their great parallel processing capacity and the success they have had in object recognition. For 3D objects, Mehta D et al. [13] developed a Hopfield network, although it is only appropriate for smooth surfaces. Wang et al. [14] specified an energy loss function that has a minimal value when the identification result is accurate, and this approach may identify several items in a single scene. The 3D ShapeNets proposed by Peng et al. [15] use a 3D CNN architecture to learn features from voxel grids for recognition purposes. The first three layers of 3D ShapeNets are convolutional layers, and the fourth layer is a fully connected layer. Considering the impact of object outlines on recognition and classification, no pooling operation is used in the network. This approach is actually a process of simulating a two-dimensional depth convolution, but the input source is changed from a picture to a voxel grid, and the two-dimensional convolution operation is changed to a three-dimensional convolution, which has achieved good results in recognition and classification. In contrast to PoseNet, which returns the 6D posture directly from RGB photos via the network, the authors [16, 17] propose to transform the 3D pose estimation issue into a classification problem by discretizing the continuous pose space. Because of its regression displacement and rotation vectors, these two quantities require hyperparameters to reconcile in the loss function. Another approach is to not directly predict the pose of the object, but to predict the pixel coordinates of the key points of the object, similar to the method proposed by Lowe D G. [3], because all the predicted values are in the 2D image, so there is no need to reconcile in the loss function. With different loss terms, the entire training process will also become more stable. Using a denoising autoencoder that employs domain randomization to train on simulated views of the 3D model, Fischler M A and Bolles R C [18] convert objects in the input picture to a vector and then determine the nearest pretrained vector that returns the proper position from the training data. Ren et al. [19] proposed a new DNN to estimate the 6D pose of an object. The paper pointed out that the method of directly returning the object pose to the image has limited accuracy, and by matching the rendered image of the object, it can be further improved. Improve accuracy, that is, given an initial pose estimate, render the synthetic RGB image to match the target input image, and then calculate a more accurate pose. Compared with traditional pose estimation methods, deep learning-based methods have better performance, which mainly relies on the powerful feature extraction capability of deep learning, which makes it suitable for pose estimation tasks. The method of global registration does not depend on the initial pose, and a commonly used method is the RANSAC algorithm proposed by Badrinarayanan et al. [20]. In each iteration of the method, the two point sets that need to be registered are first sampled, then calculated, and evaluated until the difference between the two is below a certain threshold and the iteration process terminates. This method has high requirements on the quality and accuracy of the 3D model of the object and the input point cloud and requires more expensive computing resources. References [21, 22] proposed to generate 3D bounding box candidates, first extract the point cloud of the target object, and then use a 3D convolutional network to learn voxelized features for pose estimation. Although voxel representations can efficiently encode geometric spaces, they are computationally expensive. In addition, some deep learning framework methods based on 3D point cloud can directly estimate 6D pose on 3D point cloud. The VoxelNet proposed by Gujjar H S. [23] uses 3D convolutions for feature learning on the voxelized grid of point clouds, which achieves very good results so far on the KITTI dataset. RGB-D based methods are commonly used for tasks such as indoor robot 3D object recognition, pose estimation, and grasping. The most representative of this type of methods is the LINEMOD algorithm proposed by Du et al. [24], which extracts RGB images and depth images from different perspectives to generate templates for 3D models of objects. Then use these templates to match the actual image, get the initial pose estimation, and then use the ICP algorithm to optimize. Busari et al. [25] fuse the features of the depth image on this basis, and the convolutional neural network processes the RGB image and the depth image at the same time. After obtaining the initial pose information, it is also necessary to perform postprocessing optimization on the 3D input data to obtain the final pose.

3. Method

3.1. Convolutional Neural Network Composition. Modern CNNs are mainly composed of convolutional layers, pooling layers, fully connected layers, activation functions, normalization layers, input, and output. The convolution pooling part at the front end of the network is the feature extractor, including activation function and normalization layer. The part of the backend close to the output can select active network layers according to different task types, including fully connected layers, global pooling layers, and convolutional layers. A completely connected layer is linked to the feature extractor's backend in the early stages of classification or regression, and the fully connected layer reduces the feature's dimension. However, overfitting may occur if too many parameters are included in the fully linked layer. Various components of CNN are described as follows:

(1) The convolutional layer is the core component of the CNN, which is composed of convolution kernels, and its purpose is to extract local features in the image. There is a significant reduction in the number of parameters when the convolution kernel glides across an image or feature map. Even though the receptive field of each individual convolutional kernel is modest, by stacking many convolutional layers, the receptive field of the total network may be much larger. Convolution kernels slide over an image or feature map and produce activation values based on the dot product of the convolution kernel and the current region when the CNN is forwarded. After the sliding is over, the convolutional layer outputs a new feature map.

(2) Another important component in CNNs is the pooling layer, which is a form of nonlinear downsampling. Common pooling layers are max pooling layer, average pooling layer, global max pooling layer, etc. Among them, the maximum pooling layer is the most commonly used pooling layer which divides the input into a set of nonoverlapping subregions and takes the maximum value in each subregion to represent this subregion. The purpose of using pooling layers is to obtain translation invariance, making the model focus on the presence of a feature rather than the location of the feature. In addition, the pooling layer can also reduce the resolution of the feature map, which can reduce the computational cost of the network while avoiding overfitting.

(3) Fully connected layer: high-level semantic features will be extracted after the CNN has used many layers of convolution and pooling to extract features. In the old technique, the completely linked layer serves as a "classifier." Each neuron in this layer is connected to the preceding layer. The position information in the feature map is discarded by the fully connected layer, which reduces the model learning process's parameter sensitivity.

(4) The activation function is an indispensable component in the neural network and is often used in conjunction with the convolutional layer. In order to understand complicated mapping relationships, nonlinear transformations in activation functions are utilized instead of basic linear transformations. A linear regression model is a neural network with no activation function. The commonly used activation functions are Sigmoid function, Tanh function, ReLU function, Leaky ReLU function, etc.

(5) Normalization layer: the training of CNN is a very complex process, and as the depth of the network increases, the training of the network will become more and more difficult. It is due to a number of reasons. First, if there is a slight change in the first few layers in the network, this change will gradually accumulate as the number of layers increases, thus

having a large impact. Second, if the distribution of data in a certain layer of the network changes, then the backend network of this layer needs to be relearned. During the training process, the network needs to continuously adapt to changes in the distribution of input data, and the convergence speed is affected. Third, if the distribution of the input data changes, the distribution of features at each layer in the entire network changes, a phenomenon known as internal covariate shift. To solve this problem, researchers propose batch normalization layers. In addition to the batch normalization layer, the commonly used normalization layers are group normalization layer, instance normalization layer, etc.

3.2. Optimization Method. Common optimization methods in convolutional neural networks include stochastic gradient descent (SGD), AdaGrad, Adam, etc.

(1) *SGD Algorithm*. Update network parameters:

$$\theta = \theta - \alpha \frac{1}{m} \sum_{i=1}^{m} \nabla_\theta L\left(F\left(x^{(i)};\theta\right), y^{(i)}\right), \tag{1}$$

where α is the learning rate and θ is the model parameter.

(2) *AdaGrad Algorithm*. Calculate the gradient:

$$g = \frac{1}{m} \sum_{i=1}^{m} \nabla_\theta L\left(F\left(x^{(i)};\theta\right), y^{(i)}\right). \tag{2}$$

The square of the cumulative gradient:

$$r = r + g \odot g. \tag{3}$$

Update network parameters:

$$\theta = \theta - \frac{\alpha}{\mu + \sqrt{r}} \odot g, \tag{4}$$

where α is the global learning rate, θ is the model parameter, u is a constant, and the gradient squared cumulative variable $r = 0$.

(3) *Adam Algorithm*. Update the first moment estimate:

$$s = \rho_1 s + \left(1 - \rho_1\right) g. \tag{5}$$

Update the second moment estimate:

$$r = \rho_2 r + \left(1 - \rho_1\right) g \odot g. \tag{6}$$

Correct first moment deviation:

$$\hat{s} = \frac{s}{1 - \rho_1^t}. \tag{7}$$

Correct second moment bias:

$$\hat{r} = \frac{r}{1 - \rho_2^t}. \tag{8}$$

Update network parameters:

$$\theta = \theta - \frac{\alpha \hat{s}}{\mu + \sqrt{r}} \odot g, \tag{9}$$

where ρ_1 and ρ_2 are the exponential decay rates of the moment estimates, the first-order moment variable $s = 0$, the second-order moment variable $r = 0$, and the number of time steps $t = 0$.

3.3. 3D HPE Algorithm Based on the Improved Transformer. Transformer neural network aims to solve sequence-to-sequence tasks and handle long range dependencies with ease. It is a deep learning model that uses the mechanism of attention and is composed of many self-attention layers. It differentially weights the significance of parts of the input data and processes all the input data at once by allowing parallelization, thus greatly reducing the training time. It encodes the input data as features via the attention mechanism. The input images are divided into several local patches and the representation of their relationship is calculated [26]. Transformers can be applied to various data modalities, and recent research shows that they can achieve a higher accuracy, better parameter efficiency, and computational efficiency when applied in the domain of computer vision. In this subsection, we describe the training process of our transformer-based model in detail.

3.3.1. Training Process. The camera captures the human pose in a nonspecific scene at a certain frame rate, creates a human pose dataset, and performs data enhancement. The data enhancement methods include random rotation, random scaling, and random saturation adjustment. Then randomly rotate the picture from −45° to +45°, and randomly scale the picture to 0.7~1.3 times of the original image. The implementation method of random saturation adjustment is to first set a threshold value *t*. Then randomly select a number a within (0, 1). If so, the saturation adjustment is scaled by a. If it is not satisfied, a number *b* is randomly selected within (-a, a), and the ratio of saturation adjustment is *b*+1. The two-dimensional HPE model is trained, and the image after data processing is firstly subjected to two-dimensional HPE to obtain the two-dimensional coordinates of the joint points of the human body. It specifically includes the following:

(1) The Cascaded Pyramid Network (CPN) is used for 2D HPE, and Mask R-CNN is used for human bounding box detection, where Mask R-CNN uses ResNet-101 as the backbone

(2) On the basis of the completed model, CPN selects ResNet-50 as the backbone, and the input image size is 384×288

(3) Reinitialize the last layer of the network, so that the heat map of human joint points returns to the two-dimensional joint points corresponding to the data set

(4) After training the cascade pyramid network model, input the data-enhanced image into the cascade pyramid network for 2D HPE and obtain the 2D human body joint point coordinates

In the above training process, the model hyperparameters are set to the following: iterate 10,000 times, select Adam optimizer, the number of training samples in a single batch is 16, and the learning rate uses a gradual decay strategy. The rate is 0.1. After training the improved transformer model, the two-dimensional coordinates of all human joint points are composed of a feature sequence and input into the improved transformer for 3D HPE, and the 3D coordinates of the human joint points are obtained.

3.3.2. Improved Transformer Model Training Process. Transformer is improved through switchable temporal hole network and pose graph convolution, and the improved transformer model is trained on the dataset, including the following:

(1) Switchable temporal hole network structure: the feature sequence size of the input switchable temporal hole network is (243, 34). The input feature sequence is subjected to a 1D convolution with a kernel size of 3, a dilation rate of 1, and an output channel number of 544. Then the feature goes through B blocks with residual structure. Each block first undergoes a 1-dimensional switchable time-domain hole convolution with a convolution kernel size of 3 and a hole rate of 3C. Afterwards, the feature sequence undergoes a 1D convolution with a kernel size of 1 and a dilation rate of 1. Each convolution is followed by a set of 1D batch normalization layers, ReLU activation functions, and dropout layers.

(2) Switchable temporal hole convolution: the feature sequence size of the input switchable temporal hole convolution is (H, 544). Among them, H represents the H frame image, 544 represents the number of channels, and the input feature sequence is firstly subjected to the time-domain convolution with the convolution kernel size of 3, the stride of 1, and the hole rate of 3C. The convolution kernel size is S, and the hole rate is standard convolution of 1 and self-attention. The size of the feature sequence after self-attention is $H \times H$, and then the average pooling feature size becomes (H, 1), and then the conversion factor M is obtained through 1D convolution with a convolution kernel size of 1 and SoftMax. The feature sequence K2 is obtained by multiplying M and the feature sequence after feature extraction by the time-domain hole convolution with a convolution kernel size of 3. The feature sequence K1 is obtained by multiplying (1-M) with the feature sequence obtained by feature extraction by standard convolution with a convolution kernel size of S.

(3) Self-attention mechanism: Q in the mechanism first aggregates the local feature information of joint points in the feature sequence through pose graph convolution and then performs matrix multiplication with K. Then, the weight matrix is obtained through SoftMax and finally multiplied by V to obtain the output of the graph self-attention mechanism.

(4) The relationship of the human body joint points includes the human body joint point adjacency relationship, the human body joint point symmetry relationship, and the human body joint point motion correlation relationship.

(5) There are four types of motion associations between the joints of the human body: the left wrist is connected to the right ankle, the left elbow is connected to the right knee, the right wrist is connected to the left ankle, and the right elbow is connected to the left knee.

(6) The model loss consists of two parts; one is the three-dimensional coordinate difference:

$$L_a = \sum_i^M \left\|\rho_i - \widehat{\rho}_i\right\|_2^2, \tag{10}$$

where $M = 16$, ρ_i is the three-dimensional coordinate of the i-th joint point predicted by the model, and $\widehat{\rho}_i$ represents the real value of the 3D coordinate of the i-th joint point.

The other part is the difference in the length of the bones in the symmetrical part of the human body:

$$L_b = \sum_C \left\|D_C - \widehat{D}_C\right\|_2^2, \tag{11}$$

where D_C represents the length of the C-th bone on the left, $\widehat{D}_C$ represents the length of the C-th bone on the right, and $C \in [1, 6]$.

The six symmetrical parts are the bone length difference between the neck and the left and right shoulders, the left and right shoulders and the left and right elbows, the left and right elbows and the left and right wrists, the bone length difference between the spine and the left and right buttocks, the bone length difference between the left and right hips and the left and right knees, and the left and right knees. The asymmetrical part of the human body is the difference in length between the left and right ankles. The meaning of the skeletal difference in the symmetrical part of the human body is that the length of the right wrist and the right elbow of the human body is the same as the length of the left wrist and the left elbow of the human body; that is, the ideal difference between the two should be 0, and the loss function expression is as follows:

$$L = \beta_1 L_a + \beta_2 L_b, \tag{12}$$

where β_1 and β_2 are their respective coefficients.

Finally, the transformer model designed in this paper is shown in Figure 1.

4. Experiment and Analysis

4.1. Dataset Source and Parameter Selection. A custom dataset could be used by collecting relevant images or

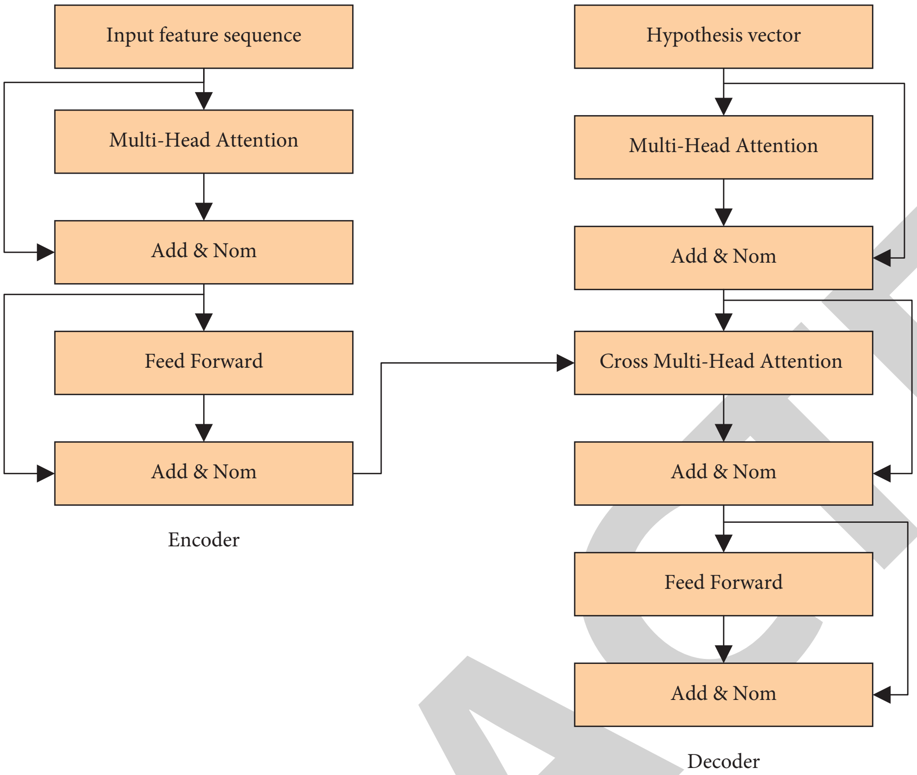

Figure 1: The transformer model designed in this paper.

using an automated tool such as that presented by [27] to create a dataset of relevant images. In our experiment, we used two state-of-the-art datasets, COCO [28] and MPII [29–31]. COCO is a large dataset provided by the Microsoft team for computer vision tasks such as HPE. COCO2017 is divided into training set, validation set, and test set. It has 200,000 images and 25,000 human labels, and each human label sentence contains 17 joints. When solving the pose estimation problem, COCO first detects the target and locates the joint points. Secondly, the evaluation criteria of pose estimation refer to the target detection criteria in this dataset. It uses object keypoint similarity (OKS) to evaluate the similarity between the ground-truth and predicted values of joints. In this paper, the overall network calculates AP (average precision) and AR (average recall) based on the OKS results. MPII is another dataset for evaluating HPE results. It contains more than 28,000 training samples and is evaluated using the PCK metric. In the data preparation stage, this paper uses DETR to detect human bounding boxes. The original image of COCO is 384×288, which is cut into blocks according to the human body bounding box and then expanded into a single-person image of the same size. Data enhancement includes the following ways: Random rotation [-45°, 45°], random scale [0.7, 1.3], and flip. The MPII data preprocessing procedure is consistent with COCO except that the image resolution is set to 384×384.

The number of encoder layers in the transformer hyperparameters has a certain influence on the experiment. Therefore, in this paper, the number of encoder layers is selected to be 6, 8, and 10 for experimental comparison. The results are shown in Figures 2–4. The selected evaluation indicators are as follows. In this classification metric, the accuracy (ACC) is used to evaluate the model when the sample distribution is balanced, which refers to the proportion of correct results in the sample.

$$ACC = \frac{TP + TN}{TP + FP + FN}, \tag{13}$$

$$PPV = \frac{TP}{TP + FP}, \tag{14}$$

where PPV is the ratio of predicted positive samples to actual positive samples.

According to the trend of the curve in the figure, among the 6-layer, 8-layer, and 10-layer encoders, the 8-layer encoder performs the best. Also under 500-epoch training, the 8-layer encoder has the highest accuracy. Under 500-epoch training, 8 layers achieve more than 90% in the ACC metric. This shows that the multihead attention mechanism used in the encoder encoding of the transformer model can better learn the relationship between pose estimates.

4.2. Model Performance Testing Experiment. We conducted model performance experiments on the two datasets and compared the results of our model with other methods. Tables 1 and 2 summarize the results.

Table 1 shows the comparison between the prediction results of this paper and other methods on the COCO test

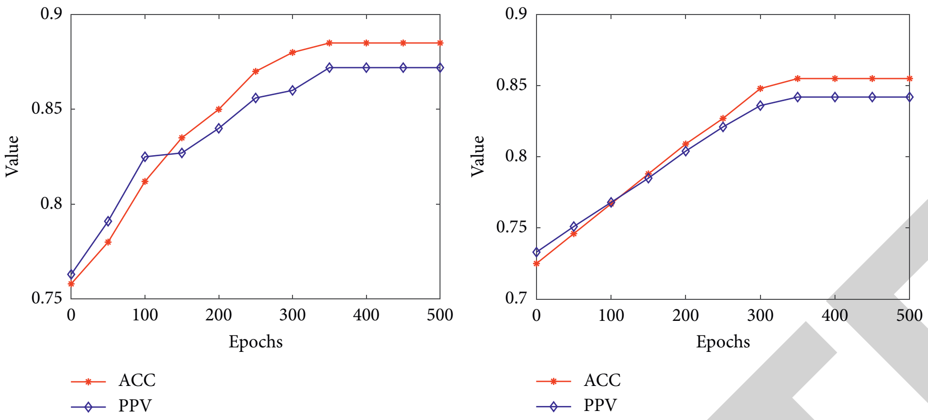

Figure 2: Indicators comparison of the model on the COCO and MPII datasets when $N = 6$.

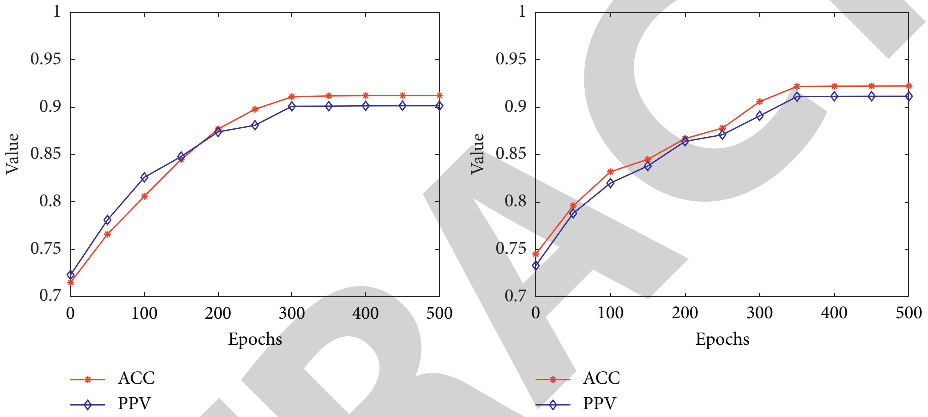

Figure 3: Indicators comparison of the model on the COCO and MPII datasets when $N = 8$.

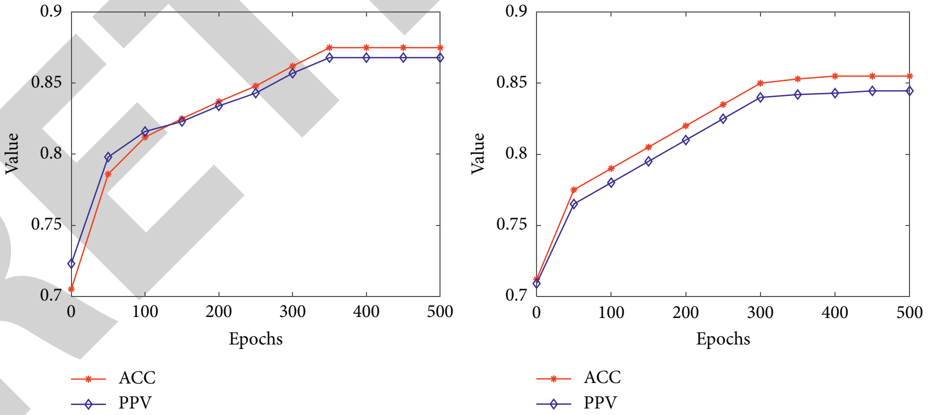

Figure 4: Indicators comparison of the model on the COCO and MPII datasets when $N = 10$.

Table 1: Prediction results of this paper and other methods on the COCO test set.

Method	Backbone	AP	AP_{50}	AP_{75}	AP_M	AP_L	AR
CVPR	ResNet-101	62.5	83.9	71.3	55.8	73.8	68.5
ECCV	ResNet-101	65.8	85.8	75.9	61.9	75.6	68.9
ICCV	ResNet-101	67.3	89.2	77.2	62.6	76.7	76.1
PRTR	ResNet-101	68.2	89.2	77.5	62.8	77.2	76.0
Our model	ResNet-101	71.9	90.5	79.8	65.5	78.8	78.1

Table 2: Prediction results of this paper and other methods on the MPII test set.

Method	Backbone	Hea	Sho	Elb	Wri	Hip	Kne	Ank
CVPR	ResNet-50	93.5	91.2	86.5	80.1	85.8	82.9	76.2
PRTR	ResNet-50	95.2	93.9	87.2	81.5	87.2	84.5	78.5
PRTR	ResNet-101	95.3	93.8	87.3	81.7	87.7	85.2	79.5
Our model	ResNet-50	95.8	93.8	88.4	82.4	88.2	86.6	80.2
Our model	ResNet-101	95.6	94.0	88.4	83.1	88.5	85.7	81.0

set. It can be seen that the AP of our method on the COCO test set is 71.9%, which is still 3.7% higher than the PRTR ratio of the same backbone network. The APs for CVPR and ECCV were only 62.5% and 65.8%, respectively. In addition, the AR of our method is 78.1%, which is 2.1% higher than PRTR.

The results on the MPII validation set are shown in Table 2 where Hea refers to head, similarly Sho refers to shoulder joint, Elb refers to elbow joint, Wri refers to wrist joint, Kne refers to knee joint, and Ank refers to ankle joint. When using ResNet-50 as the backbone network, PRTR achieved a PCKh-0.5 score of 81.5% for the wrist joint and 78.5% for the ankle joint. The scores of our method under the same conditions are 82.4% and 80.2%, respectively. When the backbone network is replaced with ResNet-101, the PCKh-0.5 scores of PRTR for wrist and ankle PRTR are 81.7% and 79.5%, respectively. The scores of our method under the same conditions are 83.1% and 81.0%, respectively. Compared with other joints, the method proposed in this paper has more advantages in the prediction results of terminal joints.

5. Conclusion

HPE is a hot research direction in computer vision. Because the image is affected by factors such as shooting angle, illumination, and surrounding environment, early HPE methods based on handcrafted features have not been able to obtain satisfactory performance. Using convolutional neural networks (CNN) to learn feature representation instead of traditional handcrafted features can achieve end-to-end optimization. Although the HPE method based on CNN has made great progress, in practical applications, it still faces some problems. On the one hand, most HPE research focuses on increasing accuracy, but it neglects the crucial balance between model speed and accuracy that is essential to HPE efficiency. Previous methods did not realize the importance of quantization error and optimization contradiction in HPE, which is a key issue to achieve high-precision HPE. These two major issues are addressed in this study by conducting research from three different angles, efficient network architecture design, model training approach, and high-precision placement. We introduce the research status of HPE at home and abroad, which provides a theoretical basis for the design of the transformer 3D HPE model. Secondly, the technical principle and optimization scheme of CNN and transformer are introduced, and a 3D HPE model based on transformer is proposed. Two well-known datasets are used to perform experiments to find the best parameters for model development. Various data enhancement techniques such as rotation, scaling, and saturation adjustments are applied and the model is trained. The experimental results show that the proposed model's prediction results are better than other methods we compared our work with.

Data Availability

The datasets used during the current study are available from the author on reasonable request.

Conflicts of Interest

The author declares that he has no conflicts of interest.

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] N. Sarafianos, B. Boteanu, B. Ionescu, and I. A. Kakadiaris, "3D Human pose estimation: a review of the literature and analysis of covariates," *Computer Vision and Image Understanding*, vol. 152, pp. 1–20, 2016.

[3] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[4] L. Sigal, "Human pose estimation," *Human Pose estimation [M]. Computer Vision: A Reference Guide*, Springer International Publishing, Cham, pp. 573–592, 2021.

[5] J. Shotton, R. Girshick, A. Fitzgibbon et al., "Efficient human pose estimation from single depth images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2821–2840, 2013.

[6] Y. LeCun, B. Boser, J. S. Denker et al., "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989.

[7] Q. Dang, J. Yin, B. Wang, and W. Zheng, "Deep learning based 2D human pose estimation: a survey," *Tsinghua Science and Technology*, vol. 24, no. 6, pp. 663–676, 2019.

[8] T. V. Marcard, G. Pons-Moll, and B. Rosenhahn, "Human pose estimation from video and IMUs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 8, pp. 1533–1547, 2016.

[9] O. Russakovsky, J. Deng, H. Su et al., "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[10] S. Li, C. Xu, and M. Xie, "A robust O(n) solution to the perspective-n-point problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1444–1450, 2012.

[11] J. J. Tompson, A. Jain, and Y. LeCun, "Joint training of a convolutional network and a graphical model for human pose estimation[J]," *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[12] D. Mehta, S. Sridhar, O. Sotnychenko et al., "VNect," *ACM Transactions on Graphics*, vol. 36, no. 4, pp. 1–14, 2017.

[13] T. W. Chen and W. C. Lin, "A neural network approach to CSG-based 3-D object recognition[J]," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 7, pp. 719–726, 1994.

[14] J. Wang, S. Tan, X. Zhen et al., "Deep 3D human pose estimation: a review," *Computer Vision and Image Understanding*, vol. 210, Article ID 103225, 2021.

[15] X. Peng, B. Sun, and K. Ali, "Exploring invariances in deep convolutional neural networks using synthetic images[J]," *CoRR*, vol. 2, no. 4, 2014.

[16] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[17] M. Jaderberg, K. Simonyan, and A. Zisserman, “Spatial transformer networks[J],” *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[18] M. A. Fischler and R. C. Bolles, “Random sample consensus,” *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[19] S. Ren, K. He, and R. Girshick, “Faster r-cnn: towards real-time object detection with region proposal networks[J],” *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[20] V. Badrinarayanan, A. Kendall, and R. Cipolla, “SegNet: a deep convolutional encoder-decoder architecture for image segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[21] Z. Liu, H. Tang, and Y. Lin, “Point-voxel cnn for efficient 3d deep learning[J],” *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[22] Y. He, G. Xia, Y. Luo et al., “DVFENet: dual-branch voxel feature extraction network for 3D object detection,” *Neurocomputing*, vol. 459, pp. 201–211, 2021.

[23] H. S. Gujjar, “A comparative study of VoxelNet and PointNet for 3D object detection in car by using KITTI benchmark,” *International Journal of Information Communication Technologies and Human Development*, vol. 10, no. 3, pp. 28–38, 2018.

[24] J. Du, C. Jiang, Z. Han, H. Zhang, S. Mumtaz, and Y. Ren, “Contract mechanism and performance analysis for data transaction in mobile social networks,” *IEEE Transactions on Network Science and Engineering*, vol. 6, no. 2, pp. 103–115, 2019.

[25] S. A. Busari, K. M. S. Huq, S. Mumtaz et al., “Generalized hybrid beamforming for vehicular connectivity using THz massive MIMO,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 9, pp. 8372–8383, 2019.

[26] A. Vaswani, N. Shazeer, N. Parmar et al., “Attention is all you need,” *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[27] A. Ali, R. Ali, A. M. Khatak, and M. S. Aslam, “Large scale image dataset construction using distributed crawling with hadoop YARN,” in *Proceedings of the Joint 10th International Conference on Soft Computing and Intelligent Systems (SCIS) and 19th International Symposium on Advanced Intelligent Systems (ISIS)*, pp. 394–399, IEEE, Toyama, Japan, 2018 Dec 5.

[28] T. Y. Lin, M. Maire, S. Belongie et al., “Microsoft coco: common objects in context,” in *Proceedings of the European conference on computer vision*, pp. 740–755, Springer, Cham, 2014 Sep 6.

[29] R. Ali, A. M. Khatak, F. Chow, and S. Lee, “A case-based meta-learning and reasoning framework for classifiers selection,” In *Proceedings of the 12th international conference on ubiquitous information management and communication*, pp. 1–6, 2018 Jan 5.

[30] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, “2d human pose estimation: new benchmark and state of the art analysis,” in *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*, pp. 3686–3693, IEEE, Columbus, OH, USA, 23-28 June 2014.

[31] R. Ali, S. Lee, and T. C. Chung, “Accurate multi-criteria decision making methodology for recommending machine learning algorithm,” *Expert Systems with Applications*, vol. 71, pp. 257–278, 2017.