Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 7316396, 14 pages
https://doi.org/10.1155/2022/7316396

Research Article

An LSTM with Differential Structure and Its Application in Action Recognition

Weifeng Chen,[1,2] **Fei Zheng**,[2,3] **Shanping Gao**,[1] **and Kai Hu**[2]

[1]*Quanzhou University of Information Engineering, Quanzhou, Fujian, China*
[2]*School of Automation, Nanjing University of Information Science & Technology, Nanjing, China*
[3]*China Telecom Ningbo Branch, Zhejiang, Ningbo, China*

Correspondence should be addressed to Weifeng Chen; cwf6426@nuist.edu.cn

Received 22 January 2022; Accepted 6 April 2022; Published 10 May 2022

Academic Editor: Saadat Hanif Dar

Because of the broad application of human action recognition technology, action recognition has always been a hot spot in computer vision research. The Long Short-Term Memory (LSTM) network is a classic action recognition algorithm, and many effective hybrid algorithms have been proposed based on basic LSTM infrastructure. Although some progress has been made in accuracy, most of those hybrid algorithms have to have more and more complex structures and deeper network levels. After analyzing the structure of the classic LSTM from the perspective of control theory, we determined that the classic LSTM could strengthen the differential characteristics of human action recognition technology to reflect the change of speed. Thus, an improved LSTM structure with an input differential characteristic module is proposed. Furthermore, in this article, we considered the influence of first-order and second-order differential on the extraction of movement pose information, that is, the influence of movement speed and acceleration on action recognition. We designed four different LSTM units with first-order and second-order differential. Moreover, the experiments were performed for the four units on three common datasets repeatedly. We found that the LSTM network with the input differential feature module proposed in this article can effectively improve action recognition accuracy and stability without deepening the complexity of the network and can be used as a new basic LSTM network architecture.

1. Introduction

With the widespread use of virtual reality technology [1], human-computer interaction, intelligent transportation [2, 3], and other fields [4] in real life, action recognition research has been rapidly developing, and action recognition occupies a pivotal position in computer vision. The goal of this research was to detect the action in video or image sequences, judge action categories, or predict further actions. At present, action recognition research methods can be divided into two categories: one is based on manual feature extraction [5–9], and the other is based on deep neural network learning features.

The method based on manual feature extraction takes the traditional machine learning method to extract features from the video, then encode the features, normalize the coding vector, train the model, and finally predict and classify the actions. Its advantage lies in extracting features according to needs, strong pertinence, and simple implementation; however, the datasets present lighting, similar actions (jogging and running), dynamic background, and other noises in action recognition [10]. These noises make the manual extraction features challenging to classify in subsequent classification tasks; therefore, the research work on action recognition based on manual feature extraction methods is currently limited—the most representative one is iDT (improved Dense Trajectories). The iDT algorithm is the most stable in this type of algorithm, but its computation speed is slow, and real-time requirements cannot be satisfied due to a large amount of calculation required.

Most of the existing network frameworks of action recognition algorithms based on deep learning [11] are developed from the convolutional neural network [12–15]. Because action recognition objects are video sequences,

they increase time-series information compared with a single image. Action recognition algorithms based on deep learning are generally used to learn the features of a time series. Long short-term memory (LSTM) is a classic action recognition algorithm used in deep networks. It is a kind of time recurrent neural network, specially designed to solve the long-term dependence problem of a general recurrent neural network (RNN). Because LSTM can process time-series information, the LSTM network is often applied in action recognition, and many effective hybrid algorithms are derived. Yue-Hei Ng et al. [16] proposed the two-stream convolutional network model combined with LSTM, reducing computational cost and learning the global video features. The two-stream convolutional network uses a convolutional neural network (CNN: AlexNet or GoogLeNet) on ImageNet to extract the image features and optical flow features of the video frames and then inputs the extracted image features and optical flow features to the LSTM network for processing to get the final result. Although the effect achieved by this network is general, it provides a new idea for the research of action recognition. Even if there is a large amount of noise in the optical flow images, the network combined with LSTM can be helpful for classification. Du et al. [17] proposed an end-to-end recurrent pose-attention network (RPAN). The RPAN combines the attention mechanism with the LSTM network to represent more detailed actions. Long et al. [18] proposed an RNN framework with multimodal keyless attention fusion. The network divides visual features (including RGB image features and optical flow features) and acoustic features into equal-length segments and inputs them to LSTM. The network's advantage is that it reduces computation cost and improves computation speed. The LSTM is applied to extract different features in this network. Song et al. [19] used skeleton information to train the LSTM and divided the network into two subnetworks: a temporal attention subnetwork and a spatial attention subnetwork. Tang et al. [20] proposed a novel coherence constrained graph (GCC) LSTM with spatio-temporal context coherence (STCC) and GCC to effectively recognize group activity, by modeling the relevant motions of individuals while suppressing the irrelevant motions. Shu et al. [21] proposed a novel hierarchical long short-term concurrent memory (H-LSTCM) to model the long-term inter-related dynamics among a group of persons for recognizing human interactions. Shu et al. [22] also proposed a novel skeleton-joint co-attention recurrent neural network (SC-RNN) to capture the spatial coherence among joints, and the temporal evolution among skeletons simultaneously on a skeleton-joint co-attention feature map in spatiotemporal space. Networks of action recognition based on deep learning are mainly based on three types: two-stream convolution network, 3D convolution network, and the LSTM network [23, 24]. With the further development of computer vision, the study of action recognition is limited to the above three networks. The attention mechanism and the NTU RGB + D skeleton dataset have also been researching hotspots in action recognition in recent years. Simultaneously, most of the existing action recognition algorithms based on deep learning are based on the classic LSTM model, which has derived many effective hybrid models.

From the development of action recognition in recent years, we can see that LSTM network is widely used in the research of action recognition. The action recognition algorithm based on the LSTM network depends on the more and more complex network framework, and the improvement of accuracy depends on the depth of the network and the number of parameters. The overcomplex hybrid networks have high requirements for machine hardware and do not improve the attention to the action fine features. At present, action recognition is mostly applied in the human-computer interaction, such as the conversion of action between a real person and a simulated digital person in somatosensory games, which pays great attention to the fineness of the action. So action recognition should pay more attention to the action posture and the extraction of action features. In order to better deal with the problems existing in the video datasets of action recognition, such as complex background, illumination transformation, and action similarity (such as walking and running), and to improve the recognition accuracy without deepening the complexity of the algorithm framework, an action recognition algorithm based on improved LSTM network is studied.

Observing the development process of action recognition research in recent years, we believe that the research work of action recognition based on the LSTM network tends to more complex mixture models, but the research results on the information of LSTM itself appear less. However, in many practical applications, the research still cares about the details of the action itself. Moreover, an overly complicated network will make recognition speed slow. In further studying the classic LSTM, we believe that if we consider the LSTM structure from control theory, the LSTM has a proportion (P) and integral (I). If we refer to the standard PID control, we can see that the classic LSTM lacks a differential (D) link. The first-order differential represents the speed of motion from the robot control, and the second-order differential represents acceleration. We can further consider adding multiple first-order or second-order enhanced input differential modules to implement different basic network models.

The contributions of this article are as follows:

(1) Improves the classical LSTM ability to capture action's speed. The idea of an input differential LSTM unit is proposed. The concept of control differential in PID is introduced into the deep learning network. It can increase the impact of time series on action recognition and consider the different speeds and accelerations of the human body. The first-order differential corresponds to the movement speed, and the second-order differential corresponds to the action acceleration. Therefore, we intend to add the differential input module in a classical LSTM structure, to enhance the capture of speed and acceleration information in motion and improve the recognition accuracy.

(2) On the basis of improving the classical LSTM architecture, this article applies it to LRCN to improve the performance of LRCN motion recognition. Based on the input gate, forget gate, and output gate of the original LSTM unit, the input of action differential (including the first-order differential and the second-order differential) is added. Furthermore, the basic LSTM algorithm with four kinds of enhanced input differential modules is designed. By testing three classic datasets, the accuracy is improved compared with the original LSTM unit, and the stability is not decreased compared with the original LSTM unit, but the training speed is weak. The enhanced input differential LSTM unit can replace the original LSTM unit and flexibly be applied in various network frameworks to realize different application scenarios. The enhanced input differential LSTM unit has a good development prospect.

This article is divided into five sections. Section 1 introduces the development process of action recognition research; Section 2 introduces related knowledge and methodology; Section 3 introduces the four related models proposed in this article; Section 4 describes the experiments performed on the three kinds of datasets to test the performance of the 4 LSTM units proposed in this article; Section 5 summarizes the work of this article.

2. Related Knowledge and Methodology

2.1. Related Knowledge

2.1.1. Recurrent Convolutional Neural Network. The recurrent neural network [25] establishes a weight connection among the input layer's neurons, hidden layer, and output layer in the neural network. The output of the network module's hidden layer at each moment depends on the information of the previous moment. The recurrent network module of RNN can learn the current moment's information and save the information of the previous time series. However, for long-time-series information, RNN is prone to the problem of gradient disappearance. Therefore, the LSTM network is proposed to solve this problem.

The LSTM network replaces the hidden layer node in the original RNN model with a memory unit [26]. The key lies in the cell state to store historical information. There are three gate structures [27] to update or delete information in the cell state through the Sigmoid function and point-by-point product operation. Figure 1 shows an LSTM unit's internal structure; from left to right are the forget gate, the input gate, and the output gate. The LSTM network can process sequence information through the cumulative linear form to avoid gradient disappearance [28] and learn long-period information. Thus, the LSTM network can be used to learn long-time sequence information.

The equation of the forget gate is

$$f_t = \sigma\left(w_f * \left[h_{t-1}, x_t\right] + b_f\right), \tag{1}$$

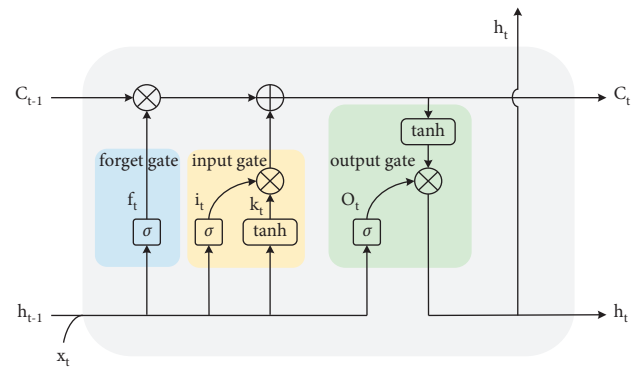

Figure 1: This is the internal structure of the long short-term memory (LSTM) network unit (basic LSTM).

where f_t is the output value of the forget gate, h_{t-1} is the output value of the last moment, x_t is the input value of the current moment, and w_f and b_f are the weight matrix and bias vector in the Sigmoid function of the forget gate, respectively. $[h_{t-1}, x_t]$ is the connection matrix of h_{t-1} and x_t.

The equations of the input gate are

$$i_t = \sigma\left(w_i * \left[h_{t-1}, x_t\right] + b_i\right), \tag{2}$$

$$k_t = \tanh\left(w_k * \left[h_{t-1}, x_t\right] + b_k\right), \tag{3}$$

where i_t and k_t are the input gate's output values, and w_i and b_i are the weight matrix and bias vector in Sigmoid function of the input gate, respectively; w_k and b_k are the weight matrix and bias vector in the tanh function of the input gate, respectively.

The equations of the output gate are

$$O_t = \sigma\left(w_O * \left[h_{t-1}, x_t\right] + b_O\right), \tag{4}$$

$$h_t = O_t * \tan\left(C_t\right), \tag{5}$$

where O_t is the output value of the output gate, w_O and b_O are the weight matrix and bias vector in the Sigmoid function of the output gate, respectively, and h_t is the output value of the current moment.

The updated cell state is

$$C_t = f_t * C_{t-1} + i_t * k_t, \tag{6}$$

where C_t is the cell state of the current moment and C_{t-1} is the cell state of the last moment.

2.1.2. PID Control. PID control is the abbreviation of proportional, integral, and differential control, which has the advantages of a simple algorithm, good robustness, and high reliability. In the control system, the PID controller constitutes the control error according to the given value and the actual output value and performs proportion, integral, and differential computations operation on the error. The three computation results are linearly combined to obtain the total control value and then control the controlled object. PID control is a linear control algorithm based on the estimation of error "past," "present," and "future" information [29]. The principle of the conventional PID control system is shown in Figure 2.

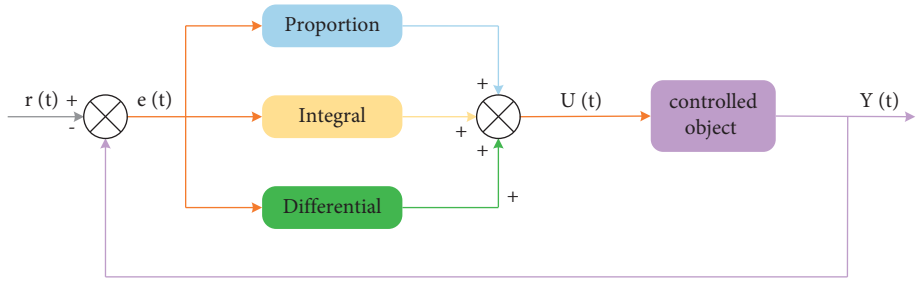

Figure 2: This is the schematic diagram of the PID control system.

Among them, $r(t)$ is the system input, $U(t)$ is the controller output, $Y(t)$ is the system output, $e(t)$ is the system error, and $e(t) = r(t) - Y(t)$.

The formula of the controller output is

$$U(t) = K_p\left[e(t) + \frac{1}{T_i}\int_0^t e(t)\mathrm{d}t + T_d\frac{\mathrm{d}e(t)}{\mathrm{d}t}\right] = K_p e(t) + K_i\int_0^t e(t)\mathrm{d}t + K_d\frac{\mathrm{d}e(t)}{\mathrm{d}t}, \tag{7}$$

where K_p is the proportional coefficient, T_i is the integral-time constant, T_d is the differential-time constant, K_i is the integral coefficient, $K_i = K_p/T_i$, and K_d is the differential coefficient, $K_d = K_p * T_d$.

Figure 2 shows that PID has three correction links: proportion, integral, and differential. The proportion link proportionally reflects the error signal $e(t)$ of the control system. Once the error occurs, the proportional controller will perform at the fastest speed to reduce error and control the "now" error. Because the adjustment function of proportional control is based on the error, it reflects PID control's rapidity. The integral link can remember the error. For the "past" error of the system, it is mainly to eliminate the steady-state error. The strength of the integral effect is mainly determined by the integral-time constant T_i. The larger the T_i is, the weaker the integral action. The integral action reflects the accuracy of PID control. The differential link can reflect the trend of the error signal (rate of change). Given the "future" error, the dynamic characteristics of the closed-loop system can be improved through advanced action, which reflects the stability of PID control.

We extract the idea of differential control in PID control. The first-order differential can increase the information capture of the LSTM unit on the action speed. The second-order differential can increase the network's information capture of the action acceleration. The improved input differential LSTM unit can improve the network's stability while improving the accuracy of the network's action recognition.

2.2. Methodology. By analyzing the classic LSTM model, we believe that the recurrent memory network retains the last video frame h_{t-1} and inputs video frame x_t, using different weights w_f and w_i to express the relationship between the frame's information. This is the relationship between the information of the LSTM frame and the current video frame. When w_f and w_i are positive, it is an integral (I) relationship between the information. When w_f and w_i are negative, it is a differential (D) relationship between the information. Simultaneously, the weight added to the current video frame's information also becomes a proportion (P) relationship. Considering that w_f and w_i are positive, when programming, the classic LSTM only contains the proportion (P) and integral (I). We believe that from the PID control, we can try to add a differential (I) relationship to the classic LSTM. From deep learning, it is also a feature enhancement idea.

From the perspective of robot kinematics, the action information features include motion limb status, posture, speed, and acceleration. Take the manipulator arm of a robot as an example, the arm's movement includes the translation of the center of mass and rotation around the center of mass. When the Newton–Euler equation analyzes the manipulator's arm, the dynamic equation is as follows:

$$\tau = M(\theta)\ddot{\theta} + V(\theta, \dot{\theta}) + G(\theta), \tag{8}$$

where $M(\theta)$ is the $n \times n$ mass matrix of the operating arm, $V(\theta, \dot{\theta})$ is the centrifugal force of $n * 1$, and the Gothic force vector, which depends on the position and speed, $G(\theta)$, is the gravity vector of $n \times 1$. $M(\theta)$ and $G(\theta)$ are complex functions of the position of all the joints θ of the manipulator's arm. $\dot{\theta}$ is the first order of angle change and represents speed. $\ddot{\theta}$ is the second-order change of angle change and represents acceleration. In control theory, the control of a robot requires first-order and second-order differential.

At present, the action recognition networks based on deep learning focus on the action posture information. So, increasing the information extraction of the action limbs' speed and acceleration can increase the network's final performance. The action speed and acceleration are the first-order and second-order differential of the posture, reflecting the posture changes trend. In this article, we introduce the differential in PID control combined with the classic LSTM

unit to realize the extraction of multiple information such as the posture, speed, and acceleration of the action.

Although the current action recognition research based on LSTM focuses on the influence of the time series on action recognition, a basic LSTM unit considers only two time series in a short period of time: the current moment and the last moment; only a part of the previous time series is retained because of the forget gate of LSTM. However, for a complete action, the action is continuous. An action cannot be completed in just two short-time sequences. A simple action (such as bowing) requires at least 3-4 time series to complete. Besides, the actions in the dataset are more complex and require more time series to complete. So, it is more effective to retain more time-series information for action recognition.

From the above ideas, this article combines the original LSTM basic unit with the differential input module to build the improved input first-order and second-order differential LSTM units. In this article, we structure a basic network framework and a multilayer LSTM to show the improved input differential network's performance. We hope that the improved input differential LSTM unit can improve the network's recognition performance in the end through the experimental results. Furthermore, we hope that the LSTM unit, combined with PID control differentiation, can capture more abundant action information. The proposed LSTM units can be flexibly applied in different networks to realize different applications.

3. LSTM Network Based on Input Differentiation

Although this article's network framework is relatively simple, it can better reflect the effect of improving the input differential LSTM unit in terms of accuracy and stability.

3.1. Improved LSTM Unit with First-Order Input Differential. Figure 3 shows that the improved first-order input differential LSTM unit adds a new input module to the original LSTM. In the mathematical model, the first-order differential of the x_t part in the differential module is $\mathrm{d}x(t)/\mathrm{dt}$. In the design of this article, since t is a fixed value and is a small value, the first-order differential of the input $\mathrm{d}x(t)/\mathrm{dt}$ is approximately as $x_t - x_{t-1}$, that is, $\mathrm{d}x(t)/\mathrm{dt} \approx x_t - x_{t-1}$; in this way, the differential can be realized, and the calculation is convenient. From the observation of basic image processing, $x_t - x_{t-1}$, the optical flow method in image processing provides the information on image change.

The state equations of the forget gate, input gate, and output gate of the LSTM unit with improved input first-order differential are shown in equations (1)–(5).

The state equations of the first-order input differential are

$$\begin{aligned} d_t &= \sigma\left(w_d * \left[h_{t-1}, x_t - x_{t-1}\right] + b_d\right), \\ e_t &= \tanh\left(w_e * \left[h_{t-1}, x_t - x_{t-1}\right] + b_e\right), \end{aligned} \tag{9}$$

where d_t is the output value of the first-order differential in Sigmoid function, and e_t is the output value of the first-order differential in tanh function, $x_t - x_{t-1}$ is the first-order input differential, w_d and b_d are the weight matrix and bias vector in Sigmoid function of the first-order input differential, respectively, and w_e and b_e are the weight matrix and bias vector in the tanh function of the first-order input differential, respectively.

The updated cell state is

$$C_t = f_t * C_{t-1} + i_t * k_t + d_t * e_t. \tag{10}$$

3.2. Improved LSTM Unit with Second-Order Input Differential. Figure 4 shows that the improved second-order input differential LSTM unit adds a second-order differential input module to the original LSTM unit. The improved input second-order differential LSTM unit is applied to the network model so that the network can extract the dual information of action features and action acceleration.

The state equations of the forget gate, input gate, and output gate of the LSTM unit with improved second-order input differential are shown in equations (1)–(5).

The state equations of the second-order input differential are:

$$\begin{aligned} d_t &= \sigma\left(w_d * \left[h_{t-1}, x_t - x_{t-2}\right] + b_d\right), \\ e_t &= \tanh\left(w_e * \left[h_{t-1}, x_t - x_{t-2}\right] + b_e\right), \end{aligned} \tag{11}$$

where d_t is the output value of the second-order differential in Sigmoid function, e_t is the output value of the second-order differential in tanh function, $x_t - x_{t-2}$ is the second-order input differential, w_d and b_d are the weight matrix and bias vector in Sigmoid function of the second-order input differential, respectively, and w_e and b_e are the weight matrix and bias vector in the tanh function of the second-order input differential, respectively.

The updated cell state is

$$C_t = f_t * C_{t-1} + i_t * k_t + d_t * e_t. \tag{12}$$

3.3. Improved LSTM Unit with Double First-Order Input Differentials. Figure 5 shows that the improved double first-order input differentials LSTM unit adds two first-order differential input modules to the original LSTM unit. The improved input double first-order differential LSTM unit is applied to the network model to extract more action characteristics and speed information.

The state equations of the forget gate, input gate, and output gate of the LSTM unit with improved double first-order input differentials are shown in equations (1)–(5).

The state equations of the double first-order input differentials are

$$\begin{aligned} d_t &= \sigma\left(w_d * \left[h_{t-1}, x_t - x_{t-1}\right] + b_d\right), \\ e_t &= \tanh\left(w_e * \left[h_{t-1}, x_t - x_{t-1}\right] + b_e\right), \\ p_t &= \sigma\left(w_p * \left[h_{t-1}, x_t - x_{t-1}\right] + b_p\right), \\ q_t &= \tanh\left(w_q * \left[h_{t-1}, x_t - x_{t-1}\right] + b_q\right), \end{aligned} \tag{13}$$

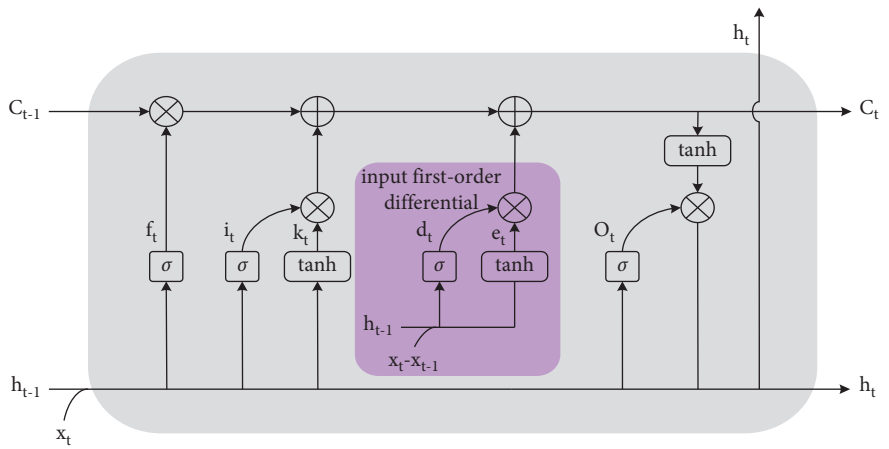

Figure 3: This is the improved LSTM unit with the first-order input differential (1st D Lstm).

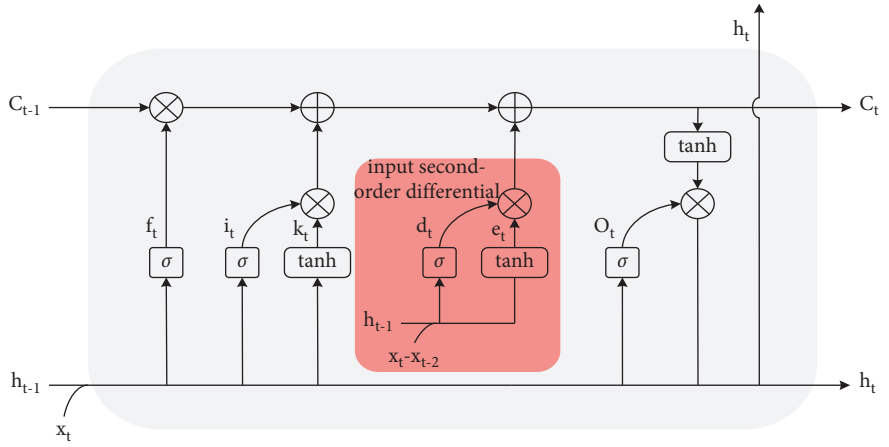

Figure 4: This is the improved LSTM unit with second-order input differential (2nd D Lstm).

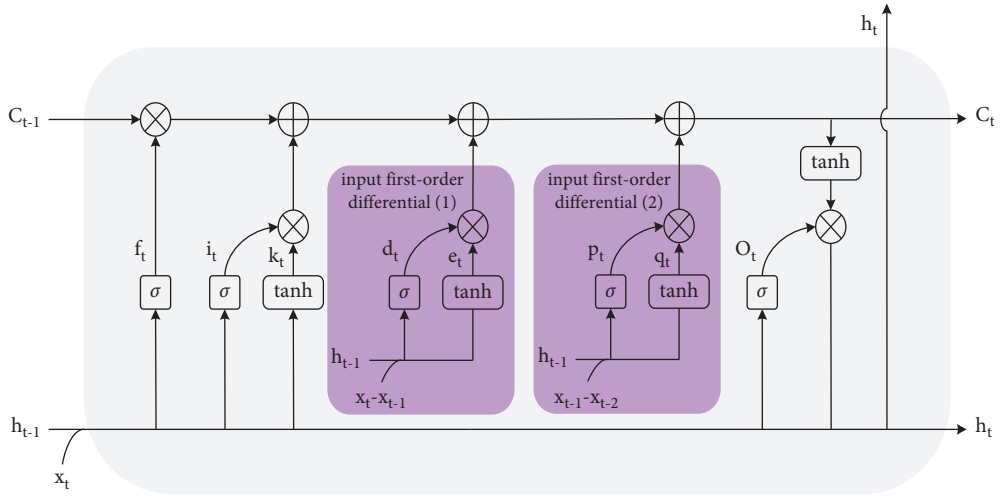

Figure 5: This is the improved LSTM unit with double first-order input differentials (double 1st D Lstm).

where d_t and p_t are the output values of the first-order differential in Sigmoid function, and e_t and q_t are the output values of the first-order differential in tanh function, $x_t - x_{t-1}$ is the first-order input differential, w_d and w_p are the weight matrices in Sigmoid function of the double first-order input differentials, b_d and b_p are bias vectors in Sigmoid function of the double first-order input differentials, w_e and w_q are the weight matrices in the tanh function of the

double first-order input differentials, and b_e and b_q are bias vectors in the tanh function of the double first-order input differentials.

The updated cell state is

$$C_t = f_t * C_{t-1} + i_t * k_t + d_t * e_t + p_t * q_t. \quad (14)$$

3.4. Improved LSTM Unit with First-Second-Order Input Differentials. As shown in Figure 6, the improved first-second-order input differential LSTM unit adds a first-order differential module and a second-order differential module to the original LSTM unit. The improved first-second-order input differential LSTM unit is applied to the network model, enabling the network to extract multiple information of motion characteristics and motion speed and acceleration.

The state equations of the forget gate, input gate, and output gate of the LSTM unit with improved first-second-order input differentials are shown in equations (1)–(5), and the state equations of first-second-order input differentials are shown in equations (15)–(18).

The state equations of the first-second-order input differentials are

$$d_t = \sigma\left(w_d * \left[h_{t-1}, x_t - x_{t-1}\right] + b_d\right), \quad (15)$$

$$e_t = \tanh\left(w_e * \left[h_{t-1}, x_t - x_{t-1}\right] + b_e\right), \quad (16)$$

$$p_t = \sigma\left(w_p * \left[h_{t-1}, x_t - x_{t-2}\right] + b_p\right), \quad (17)$$

$$q_t = \tanh\left(w_q * \left[h_{t-1}, x_t - x_{t-2}\right] + b_q\right), \quad (18)$$

where d_t is the output value of the first-order differential in Sigmoid function, e_t is the output value of the first-order differential in tanh function, $x_t - x_{t-1}$ is the first-order input differential, p_t is the output value of the second-order differential in Sigmoid function, q_t is the output value of the second-order differential in tanh function, $x_t - x_{t-2}$ is the second-order input differential, w_d and w_p are the weight matrices in Sigmoid function of the first-second-order input differentials, b_d and b_p are the bias vectors in Sigmoid function of the first-second-order input differentials, w_e and w_q are the weight matrices in the tanh function of the first-second-order input differentials, and b_e and b_q are bias vectors in the tanh function of the first-second-order input differentials.

The updated cell state is

$$C_t = f_t * C_{t-1} + i_t * k_t + d_t * e_t + p_t * q_t. \quad (19)$$

4. Experiment

4.1. Datasets. Research teams, both overseas and domestic, usually use human action datasets in algorithm training to detect the algorithm's accuracy and robustness. The dataset has at least the following two important functions:

(1) The researchers need not care about the process of collection and pretreatment.

(2) Ability to detect and compare different performances of different algorithms under the same standard.

The KTH dataset [30] was released in 2004. The KTH dataset includes six kinds of actions (walking slowly, jogging, running, boxing, waving, and clapping) performed by 25 people in 4 different scenes. The dataset has 2391 video samples and includes scale transformation, clothing transformation, and lighting transformation. However, the shooting camera is fixed and the background is relatively single.

The Weizmann dataset [31] was released in 2005 and includes nine people completing ten kinds of actions (bending, stretching, high jump, jumping, running, standing, hopping, walking, waving1, and waving2). In addition to category tags, the dataset contains silhouettes of people in the foreground and background sequences to facilitate background extraction. However, the dataset has a fixed perspective and simple backgrounds.

The above two datasets were released early. The citation rate of the datasets in the traditional methods of action recognition is high, which significantly promotes action recognition for the future. However, with the rapid development of action recognition, there are shortcomings: the background is simple, the perspective is fixed, and each video has only one person. The above two datasets already cannot satisfy real action recognition requirements, so now they are rarely used.

The Hollywood2 dataset [32] was released in 2009. The video data in the dataset were collected from Hollywood movies. There are 3669 video clips in total, including 12 action categories (answering the phone, eating, driving, etc.) extracted from 69 movies and 10 scenes (outdoor, shopping mall, kitchen, etc.). The dataset is close to real situations.

The University of Central Florida released the UCF-101 dataset [33] in 2012. The dataset samples include various action samples collected from TV stations and video samples saved from the video website YouTube. There are 13,320 videos, including five types of actions (human-object interaction, human-human interaction, limb movements, body movement, and playing musical instruments), and 101 class-specific small actions. The dataset has many samples and rich action categories and can train the algorithm well, so it is widely used.

Brown University released the HMDB-51 dataset [34] in 2011. The video samples come from the video clips of the movie and video website YouTube. There are 51 types of sample actions and 6849 videos in total. Each type of sample action in the dataset contains at least 101 videos.

The UCF-101 dataset and the HMDB-51 dataset have many action data types and a wide range of actions and are more classic in action recognition. The scenes in the Hollywood2 dataset are more complex and closer to real life. To comprehensively reflect the improved LSTM unit's performance proposed in this article, we adopted three databases, including UCF-10, 1 HMDB-51, and Hollywood2, for training and testing. Furthermore, the improved LSTM units were tested and improved performance in the above three databases.

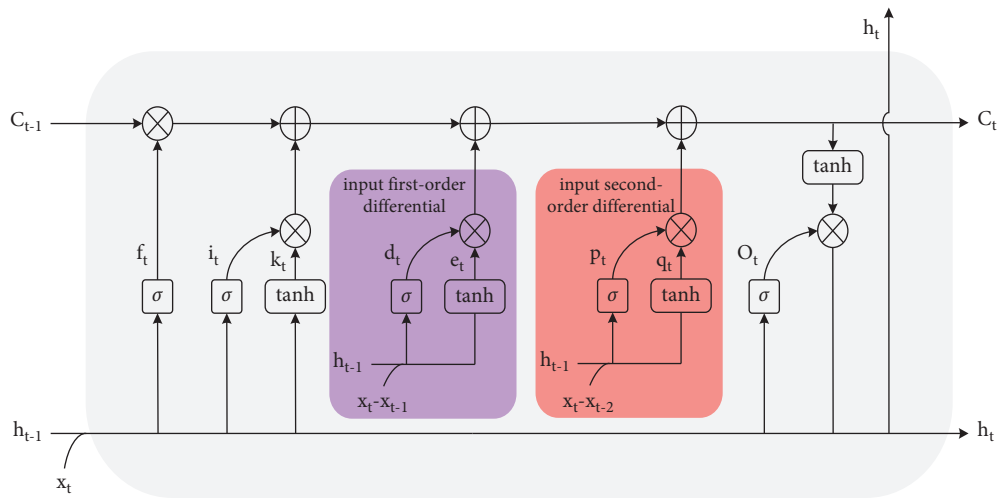

Figure 6: This is the improved LSTM unit with first-second-order input differentials (1st + 2nd D Lstm).

4.2. Experimental Method. To more intuitively and effectively test the effect of the improved input differential LSTM unit proposed in this article on action recognition, the experiments adopted a relatively simple network framework model: long-term recurrent convolutional networks (LRCNs) [35]. In future research, the four different input differential LSTM units proposed in this article can be directly used instead of the original basic LSTM units or directly replaced into a more complex network model containing LSTM units to achieve better application performance.

The LRCN directly connects the LSTM model to the convolutional neural network and simultaneously learns temporal and spatial features. The LRCN network framework is shown in Figure 7. The model converts the video data in the dataset into frame images and then uses the pretrained CNN network to extract frame images' features. Moreover, the LRCN inputs the extracted features to the improved input differential LSTM network to extract the time sequence information and finally classifies the results by SoftMax.

This article uses the method in Donahue's work [33]. The method uses the convolutional network to extract the spatial features and the LSTM network to extract the temporal features. However, the method in this article is slightly different from the original text. In the CNN feature extraction step, we adopted InceptionV3 to extract more accurate frame image features. The InceptionV3 requires little computation close but has high performance. In the step of extracting time sequence information by LSTM network, the number of network layers is customized according to computer performance and recognition accuracy requirements. Moreover, the different orders of the input differential LSTM units proposed in this article were adopted in the LRCN.

The LSTM units (discussed in Sections 2.1 and 3.1–3.4) were applied to the network model framework in Figure 7. The improved LSTM units were evaluated from the three indexes of accuracy, loss, and standard deviation. To better reflect the improved LSTM units' performance, the experiments were carried out on three datasets of HMDB-51, UCF-101, and Hollywood2, respectively. The experiments use only a single variable of the LSTM unit. The input data model, training parameters, and other parameters were consistent. The batch_size is 32, the hidden layers' parameter is 1024, the full connection layers' parameter is 512, and the loss function is the classic cross-entropy function. Moreover, the optimizer is Adadelta optimizer, the learning rate is 0.001, and the decay rate is 0.95. In LRCN, we adopted 5 levels, and each level has 1024 LSTMs to build its structure. All works are end-to-end training and end-to-end models.

The recording was as follows: recording the original LSTM unit as basic Lstm; recording the first-order input differential LSTM unit as 1st D Lstm (the model in Section 3.1); recording the second-order input differential LSTM unit as 2nd D Lstm (the model in Section 3.2); recording the double first-order input differentials LSTM as double 1st D Lstm (the model in Section 3.3); and recording the first-second-order input differentials LSTM unit as 1st + 2nd D Lstm (the model in Section 3.4).

The assessment method used the direct hold-out method. To avoid the influence of the other bias introduced by the data division process on the result and increase the final evaluation result's fidelity, the training set and the testing set were divided equally at each type of action in every dataset in the experiment. The training set accounts for 70% of the total dataset, and the testing set accounts for 30% of the total dataset. Simultaneously, to make the results more stable and reliable, this article uses multiple hold-out to take the average value as the experiment's final evaluation results. Each LSTM unit uses the hold-out method described above to divide the dataset and then conduct the experiment. After an experiment concluded, the dataset was redivided, and the experiment was performed again and repeated. The experiments were performed using three datasets of five different LSTM units; each was repeated three times. At last, the average accuracy of three experimental results is the result of the LSTM unit.

Our experiment's hardware configuration was an Intel I7-9700K CPU, two Nvidia GeForce GTX2080Ti graphics

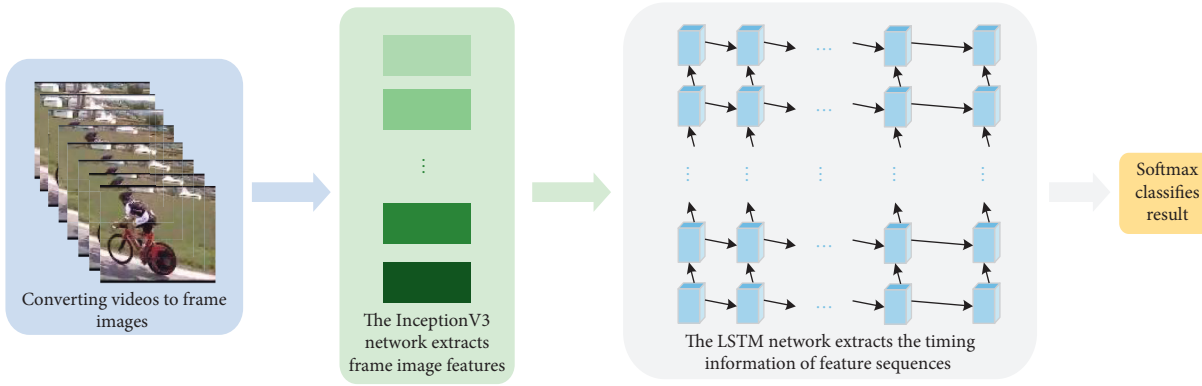

Figure 7: This is the LSTM network framework based on improved input differentiation.

cards, 4 ∗ 16 G total 64 GB memory. The software environment was configured as Ubuntu 16.04, CUDA 8.0, Cudnn 6.0 for CUDA 8.0, TensorFlow 1.4, and python 3.5.

4.3. Experimental Results and Analysis

4.3.1. Performance Experiments of Five Models in This Article on the HMDB-51 Dataset. Figures 8(a) and 8(b) show the comparison graphs of accuracy and loss of action recognition applied to five different LSMT units on the HMDB-51 dataset.

Figure 8(a) shows that on the HMDB-51 dataset, the accuracy of basic Lstm is 39.99%, the accuracy of 1st D Lstm is 41.34%, the accuracy of 2nd D Lstm is 41.44%, the accuracy of double 1st D Lstm is 42.27%, and the accuracy of 1st + 2nd D Lstm is 43.30%.

The above experimental data show that in the HMDB-51 dataset, the four different LSTM units proposed in this article have different degrees of improvement in the accuracy of action recognition than the original LSTM unit. However, the train-step is delayed to some extent when the accuracy reaches a stable level. Although the basic LSTM unit's loss is low overall, sometimes there is a step phenomenon in the loss. The loss of the 1st D LSTM unit is slightly higher than that of the basic LSMT unit. Compared with the loss of the above two LSTM units, the loss of double 1st D LSTM unit, 2nd D LSTM unit, and 1st + 2nd D LSTM unit is higher. In general, in the HMDB-51 dataset, the LSTM algorithm with enhanced input differential features is improved in accuracy compared with the classic LSTM algorithm without input differential features.

4.3.2. Performance Experiments of Five Models in This Article on the UCF-101 Dataset. Figures 9(a) and 9(b) show the comparison graphs of accuracy and loss of action recognition applied to five different LSMT units on the UCF-101 dataset.

Figure 9(a) shows that, on the UCF-101 dataset, the accuracy of basic Lstm is 71.15%, the accuracy of 1st D Lstm is 79.88%, the accuracy of 2nd D Lstm is 73.42%, the accuracy of double 1st D Lstm is 71.99%, and the accuracy of 1st + 2nd D Lstm is 72.67%.

From the above experimental data, it can be concluded that on the UCF-101 dataset, the 1st LSTM unit has the highest accuracy, and the overall loss is low, but there are still higher steps. Besides, with the superposition of training steps, there is a fluctuation in the loss. The other three LSTM units' networks' accuracy also improved compared to the original LSTM unit. In general, in the UCF-101 dataset, the LSTM algorithm with enhanced input differential features is improved in accuracy compared to the classical LSTM algorithm without input differential features.

4.3.3. Performance Experiments of Five Models in This Article on the Hollywood2 Dataset. Figures 10(a) and 10(b) show the comparison graphs of accuracy and loss of action recognition applied to five different LSMT units on the Hollywood2 dataset.

Figure 10(a) shows that on the Hollywood2 dataset, the accuracy of basic Lstm is 46.49%, the accuracy of 1st D Lstm is 47.85%, the accuracy of 2nd D Lstm is 47.89%, the accuracy of double 1st D Lstm is 46.54%, and the accuracy of 1st + 2nd D Lstm is 47%. The videos in the Hollywood2 dataset are relatively long, and the scenes are complicated. There are some interference actions except for tags in a video. Maybe for the above reason, 1st D Lstm performed better on the Hollywood2 dataset.

In general, in the Hollywood2 dataset, the LSTM algorithm with enhanced input differential features improved accuracy compared with the classical LSTM algorithm without input differential features. Table 1 shows the ablation experimental data (including mean accuracy and deviation) on 3 sets of datasets, and the bolded data are the highest.

Other researchers [36] used the LRCN model to perform action recognition, obtaining an accuracy of 38.8% and 68.3% on the HMDB-51 and UCF-101 datasets, respectively. The results are similar to the experimental results in this article; thus, this article's experimental data are credible. Through experiments, we found that different classes of LSTM differential units have different accuracy on different datasets. Compared with the original LSTM unit, the accuracy of four differential LSTM units had specific improvements. For the dataset with only a single action in videos, the first-order input differential LSTM unit might

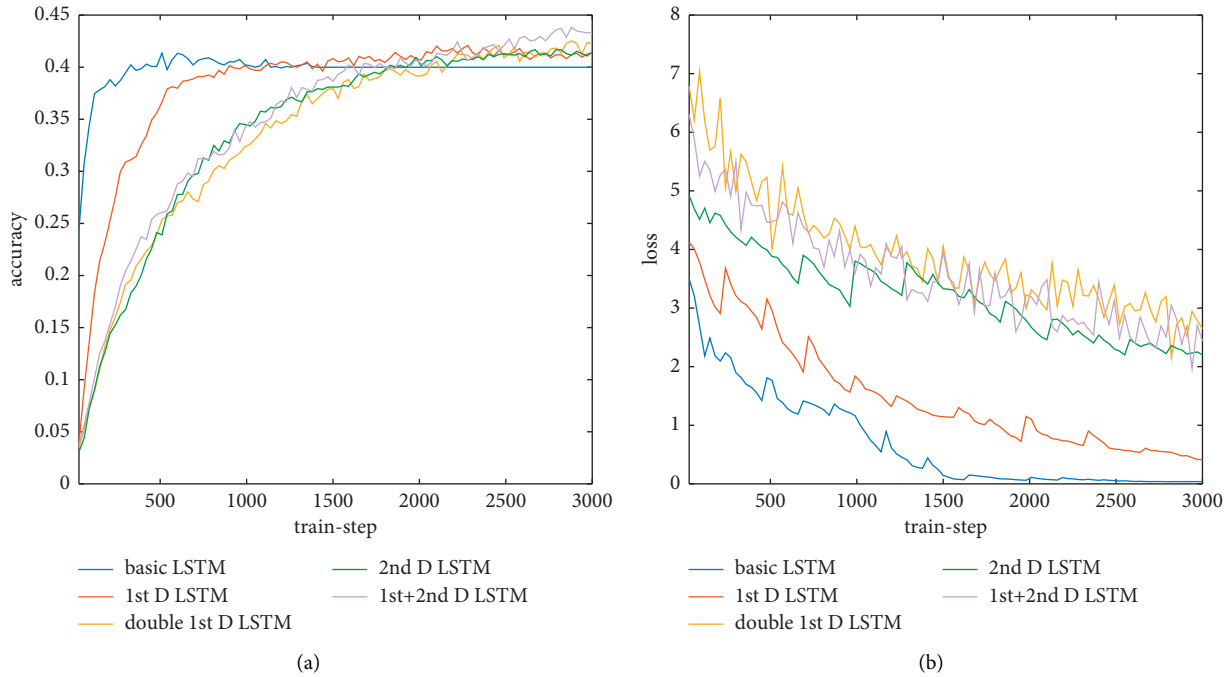

Figure 8: The comparison graph of accuracy and loss applied to five different LSMT units on the HMDB-51 dataset. (a) Accuracy and (b) Loss.

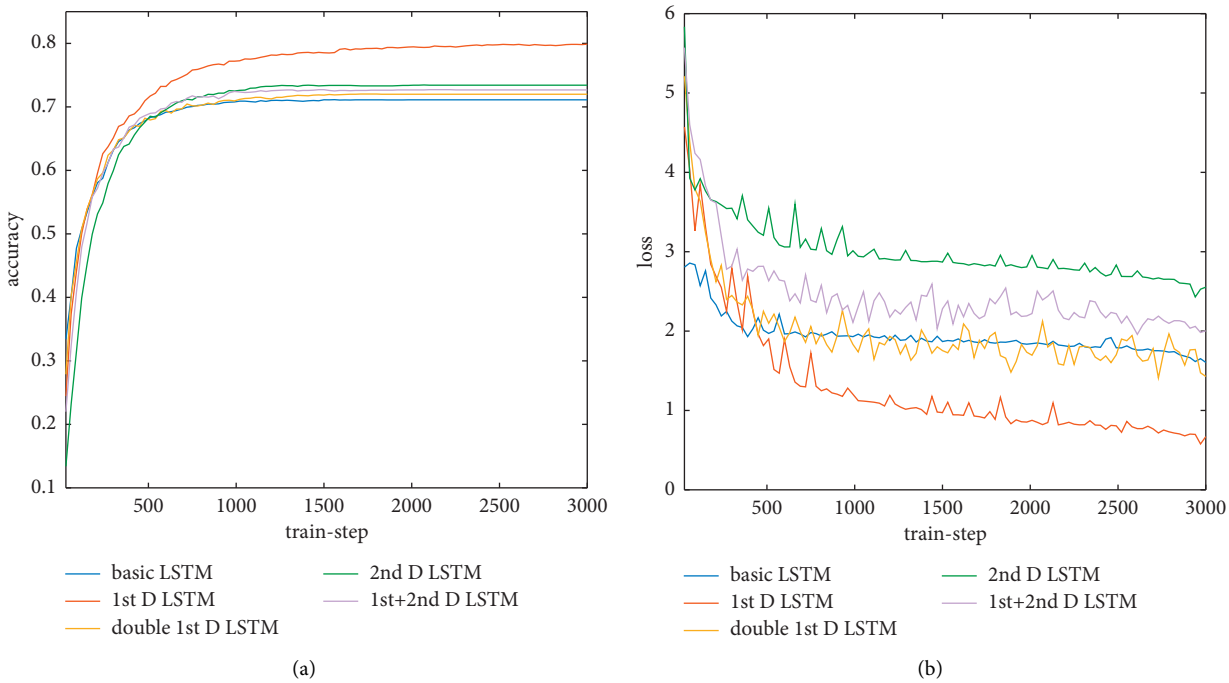

Figure 9: The comparison graph of accuracy and loss applied to five different LSMT units on the UCF-101 dataset. (a) Accuracy and (b) Loss.

work better. For the dataset with complex scenes and many other action interferences, the second-order input differential LSTM unit might work better. The four input differential LSTM units proposed in this article need to be studied further regarding loss functions. Using different optimizers or redefining loss functions may be the approach required to achieve an optimal model.

At the same time, we noticed that, in Figures 8(b), 9(b), 10(b), all D LSTMs' loss function fluctuates; we thought that, according to control theory, the differential elements easily introduce high-frequency measurement noise, which made all D LSTMs' loss more volatile than classical LSTM and 2nd D LSTMs' loss more volatile than 1st D LSTMs' loss, which are the shortcomings of all D LSTMs.

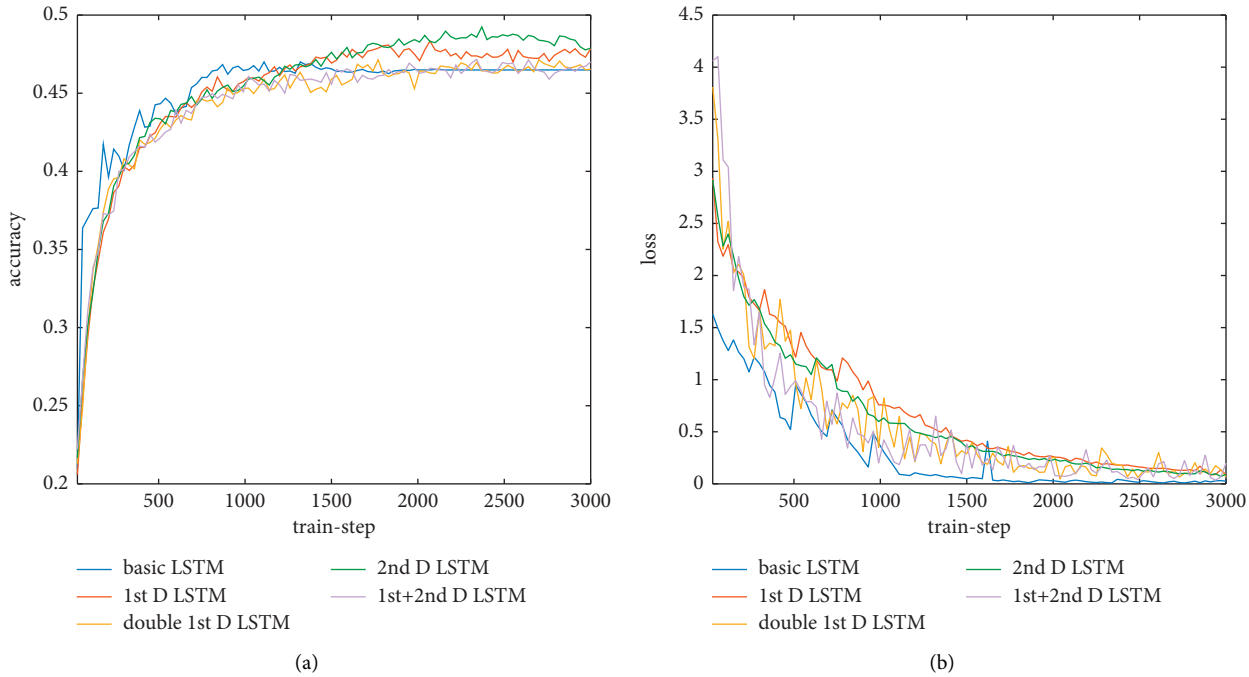

Figure 10: The comparison graph of accuracy and loss applied to five different LSMT units on the Hollywood2 dataset. (a) Accuracy and (b) Loss.

Table 1: The accuracy and standard deviation of different LSTM units on the dataset.

	HMDB-51	UCF-101	Hollywood2
Basic D Lstm	39.99% ± 1.31%	71.15% ± 0.37%	46.49% ± 0.68%
1st D Lstm	41.34% ± 0.83%	79.88% ± 0.38%	47.85% ± 0.99%
2nd D Lstm	41.44% ± 0.91%	73.42% ± 0.43%	47.89% ± 1.23%
Double 1st D Lstm	42.24% ± 1.05%	71.99% ± 0.61%	46.54% ± 0.74%
1st + 2nd D Lstm	43.30% ± 1.46%	72.67% ± 0.87%	47.00% ± 1.05%

4.4. Accuracy Comparison of Deep Learning Action Recognition Algorithms. In order to further verify the four input differential LSTM units proposed in Section 3, the improved differential LSTM units are compared with other deep learning algorithms. And experiments are carried out on UCF-101 and HMDB-51 datasets commonly used in deep learning. Table 2 is the result of a comparison experiment, and the bolded data are the highest. From Table 2, we can see that LRCN combined with 1st D LSTM is the best, and LRCN combined with 1st + 2nd D LSTM is the best.

In this section, the accuracy of two-stream convolutional network, LRCN network with attention mechanism, and LRCN network with BiLSTM is compared with the accuracy of the improved differential LSTM unit. Through experiments, it is found that the accuracy of the 1st D LSTM unit is the highest on the UCF-101 dataset and that of the 1st + 2nd D LSTM unit on the HMDB-51 dataset is the highest. The 1st D LSTM unit can better deal with features with short completion time and large category gap. The video in the UCF-101 dataset only has label actions, and there are great differences among 101 types of actions. The HMDB-51 dataset has more irrelevant actions than UCF-101 dataset. The 1st + 2nd D LSTM unit can handle both long- and short-time sequence features at the same time, so it can deal with noise actions better.

Table 2: The accuracy comparison of various deep learning algorithms on UCF-101 and HMDB-51 datasets.

		UCF-101 (%)	HMBD-51 (%)
Two-stream convolutional network [37]		73.00	40.50
LRCN	Basic LSTM	71.15	39.99
	1st D LSTM	**79.88**	41.34
	2nd D LSTM	73.42	41.44
	Double 1st D LSTM	71.99	42.24
	1st + 2nd D LSTM	72.67	**43.30**
	LSTM + attention	72.40	41.50
	BiLSTM [31]	70.00	39.81

4.5. The Stability Experiments of Five LSTM Models. In our research, the networks of five different LSTM units were tested on three datasets, repeated three times for each dataset. The average accuracy and standard deviation of the final stable results were calculated. The standard deviation can reflect the accuracy dispersion degree of the LSTM unit on the corresponding dataset. Figure 11 and Table 1 show that each LSTM unit's standard deviations applied to different datasets are small; therefore, in terms of stability, the four different input differential LSTM units proposed in this

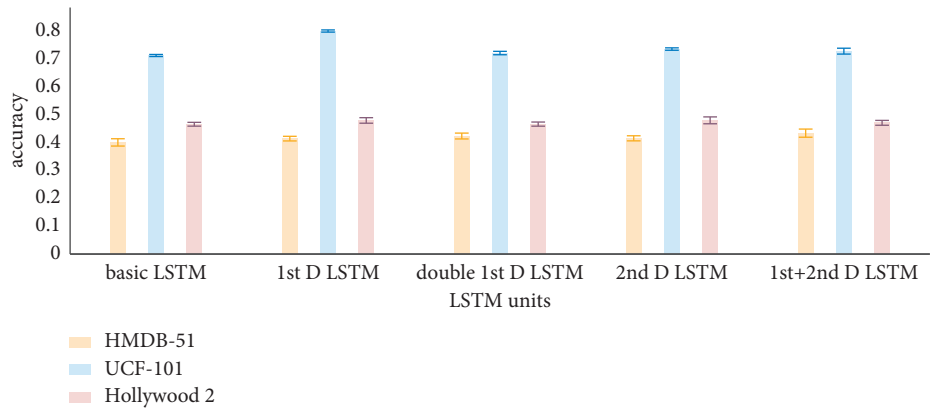

Figure 11: The comparison chart of accuracy and standard deviation of five different LSMT units.

Table 3: This is the frames per second (FPS) of different LSTM units trained on different datasets.

	HMDB-51	UCF-101	Hollywood2
Basic Lstm	61	64	32
1st D Lstm	36	36	25
2nd D Lstm	35	36	25
Double 1st D Lstm	18	20	11
1st + 2nd D Lstm	18	18	11

article have good stability in various datasets. Compared to the classic LSTM without input differential features link, the stability is not much different.

4.6. Algorithms Execution Time Experiments. The frames per second (FPS) evaluation index is a definition in the image field. Image detection and recognition generally refer to the number of images that can be processed in one second. In this experiment, FPS refers to the number of video frames that can be processed in one second.

Table 3 shows that in the process of training data, on the whole, the original LSTM unit processes more image frames in one second. As the amount of data processed by the network doubles, the input differential LSTM units used in the HMDB-51 and UCF-101 datasets show slower data processing. However, in the Hollywood2 dataset, it is equivalent to the original LSTM unit. The speed of processing data in different LSTM units may be affected by the video content's complexity. In terms of training time, compared with the classical LSTM algorithm without input differential features, the four methods proposed in this article are all inferior.

When the trained model parameters were used for recognition on different datasets, the original LSTM unit's recognition speed and the four LSTM units proposed in this article on a limited number of datasets are similar, which is roughly around 180 frames per second. However, overall, the original LSTM unit's recognition speed is 4 to 7 frames per second faster than the proposed four LSTM units proposed in this article.

5. Conclusion and Prospect

Human action recognition has more application requirements today and has received significant attention from researchers in related fields [38]. In this study, we combined the differentiation idea in PID control with the LSTM unit in the deep learning network and proposed four kinds of LSTM units with input differentiation, which increases the influence of information difference in time series on action recognition. Compared with the complex hybrid models, the differential LSTM unit can maintain the simplicity of the network structure and improve the recognition accuracy, so that it can be better applied to the real use scene. Due to the different habits and speeds of different characters in the dataset, the input differential LSTM units proposed in this article can pay attention to body movement speed to increase the characteristic information of actions in the time series. The experiments prove that the four different LSTM units proposed in this article have different degrees of improvement in action recognition accuracy compared with the original LSTM units. According to the video's length and the video's actions, different differential units have different performances in each dataset. Compared with other action recognition algorithms based on deep learning, the input differential LSTM unit has advantages in recognition accuracy, and it can be used in application scenarios such as attitude estimation and image caption generation.

In summary, since most of the human actions in current action recognition datasets involve short-length videos of human actions, the accuracy of the LSTM with the first-order input differential is higher in the action recognition

network. We used a simple network structure and network parameters to reflect the input differential LSTM unit's action recognition performance. Although the input differential LSTM unit intuitively reflects good accuracy, the loss function's processing is not detailed enough and may be optimized in future applications.

In general, the first-order/second-order input differential LSTM unit proposed in this article achieved good results in action recognition. Compared with the original LSTM unit, it has improved accuracy while maintaining stability, although its training speed is weak. The proposed unit can replace the original LSTM unit, can be flexibly applied in various network frameworks to realize different application scenarios, and has a good development prospect.

Data Availability

The code used to support the findings of this study are available from the corresponding author upon request. The data are from the open dataset of HMDB-51 (UCF-101" title="https://serre-lab.clps.brown.edu/resource/hmdb-a-large-human-motion-database/), UCF-101(https://serre-lab.clps.brown.edu/resource/hmdb-a-large-human-motion-database/), UCF-101 (http://www.crcv.ucf.edu/data/UCF101.php), and Hollywood2 (http://www.di.ens.fr/~laptev/actions/hollywood2/).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

All authors drafted the manuscript, and read and approved the final manuscript.

References

[1] E. A. Suma, D. M. Krum, B. Lange, S. Koenig, S. Rizzo, and A. Bolas, "Adapting user interfaces for gestural interaction with the flexible action and articulated skeleton toolkit," *Computers & Graphics*, vol. 37, no. 3, pp. 193–201, 2013.

[2] C. Chen, B. Liu, S. Wan, P. Qiao, and P. Pei, "An edge Traffic flow detection Scheme based on deep learning in an intelligent transportation system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1840–1852, 2021.

[3] X. Gejing and L. Yang, "Research on the impact of Internet evolution on accounting information system based on data Mining," *Journal of Physics: Conference Series*, vol. 1345, no. 5, Article ID 052055, 2019.

[4] C. Chen, Q. Hui, W. Xie, S. Wan, S. Zhou, and Y. Pei, "Convolutional Neural Networks for forecasting flood process in Internet-of-Things enabled smart city," *Computer Networks*, vol. 186, Article ID 107744, 2021.

[5] X. Yang and Y. Tian, "Action recognition using super Sparse coding vector with spatio-temporal Awareness," in *Proceedings of the European Conference on Computer Vision*, pp. 727–741, Switzerland, September 2014.

[6] X. Peng, C. Zou, Y. Qiao, and Q. Peng, "Action recognition with Stacked Fisher vectors," in *Proceedings of the European Conference on Computer Vision*, pp. 581–595, Springer, Cham, September 2014.

[7] X. Peng, L. Wang, X. Wang, and Y. Qiao, "Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice," *Computer Vision and Image Understanding*, vol. 150, pp. 109–125, 2016.

[8] R. Arandjelovic and A. Zisserman, "All about VLAD," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 1578–1585, IEEE, Portland, OR, USA, June 2013.

[9] I. C. Duta, B. Ionescu, K. Aizawa, and N. Sebe, "Spatio-temporal VLAD encoding for human action recognition in videos," in *Proceedings of the nternational Conference on Multimedia Modeling*, pp. 365–378, Reykjavík, Iceland, January 2017.

[10] H. Zhu, C. Zhu, and Z. Xu, "Research advanves on human activity recognition datasets," *Acta Automatica Sinica*, vol. 44, pp. 978–1004, 2018.

[11] Q. Wang and K. Chen, "Multi-label zero-shot human action recognition via joint latent ranking embedding," *Neural Networks*, vol. 122, pp. 1–23, 2020.

[12] M. Xia, W. Song, X. Sun, J. Liu, T. Ye, and Y. Xu, "Weighted Densely connected convolutional networks for Reinforcement learning," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 34, no. 04, Article ID 2052001, 2020.

[13] M. Xia, W. A. Liu, K. Wang, X. Zhang, and Y. Xu, "Non-intrusive load disaggregation based on deep dilated residual network," *Electric Power Systems Research*, vol. 170, pp. 277–285, 2019.

[14] M. Xia, J. Qian, X. Zhang, J. Liu, and Y. Xu, "River Segmentation based on Separable attention residual network," *Journal of Applied Remote Sensing*, vol. 14, no. 03, p. 1, 2019.

[15] M. Xia, X. Zhang, W. a. Liu, L. Weng, and Y. Xu, "Multi-stage feature Constraints learning for Age estimation," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2417–2428, 2020.

[16] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: deep networks for video classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4694–4702, IEEE, Boston, MA, USA, June 2015.

[17] W. Du, Y. Wang, and Y. Qiao, "Rpan: an end-to-end recurrent pose-attention network for action recognition in videos," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3725–3734, IEEE, Venice, Italy, October 2017.

[18] X. Long, C. Gan, G. De Melo et al., "Multimodal keyless attention fusion for video classification," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, USA, February 2018.

[19] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu, "An end-to-end spatio-temporal attention model for human action recognition from skeleton data," in *Proceedings of the Thirty-first AAAI conference on artificial intelligence*, San Francisco California USA, February 2017.

[20] J. Tang, X. Shu, R. Yan, and L. Zhang, "Coherence Constrained graph LSTM for group Activity recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 2, pp. 636–647, 2022.

[21] X. Shu, J. Tang, G.-J. Qi, W. Liu, and J. Yang, "Hierarchical long short-term Concurrent memory for human interaction recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 3, pp. 1110–1118, 2021.

[22] X. Shu, L. Zhang, G.-J. Qi, W. Liu, and J. Tang, "Spatio-temporal Co-attention recurrent neural networks for human-

skeleton motion prediction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 6, pp. 3300–3315, 2022.

[23] K. Hu, Y. Ding, J. Jin, L. Weng, and M. Xia, "Skeleton motion recognition based on multi-scale deep spatio-temporal features," *Applied Sciences*, vol. 12, no. 3, Article ID 1028, 2022.

[24] K. Hu, F. Zheng, L. Weng, Y. Ding, and J. Jin, "Action recognition algorithm of spatio-temporal differential LSTM based on feature enhancement," *Applied Sciences*, vol. 11, no. 17, p. 7876, 2021.

[25] J. Donahue, L. Anne Hendricks, S. Guadarrama et al., "Long-term recurrent convolutional networks for visual recognition and Description," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2625–2634, Boston, MA, USA, June 2015.

[26] M. X. Jiang, C. Deng, Z. G. Pan, L. F. Wang, and X. Sun, "Multiobject Tracking in videos based on LSTM and deep Reinforcement learning," *Complexity*, vol. 2018, Article ID 4695890, 12 pages, 2018.

[27] M. Xia, W. a. Liu, K. Wang, W. Song, C. Chen, and Y. Li, "Non-intrusive load disaggregation based on composite deep long short-term memory network," *Expert Systems with Applications*, vol. 160, Article ID 113669, 2020.

[28] W. Lu, J. Li, Y. Li, A. Sun, and J. Wang, "A CNN-LSTM-Based model to Forecast Stock Prices," *Complexity*, vol. 2020, Article ID 6622927, 10 pages, 2020.

[29] H. Wang, y. Luo, W. An, Q. Sun, J. Xu, and L. Zhang, "PID controller-based stochastic optimization acceleration for deep neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 12, pp. 5079–5091, 2020.

[30] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: a local SVM approach," in *Proceedings of the 17th International Conference on Pattern Recognition*, pp. 32–36, Cambridge, UK, August 2004.

[31] B. Moshe, G. Lena, and S. Eli, "Actions as Space-time Shapes," in *Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05)*, pp. 1395–1402, Beijing, China, October 2005.

[32] M. Marszalek, I. Laptev, and C. Schmid, "Actions in context," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2929–2936, Miami, USA, June 2009.

[33] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: a dataset of 101 human actions classes from videos in the wild," 2012, https://arxiv.org/abs/1212.0402.

[34] H. Kuehne, H. Jhuang, and E. Garrot, "HMDB: a large video database for human motion recognition," in *Proceedings of the International Conference on Computer Vision*, pp. 2556–2563, Barcelona, Spain, November 2011.

[35] J. Donahue, L. A. Hendricks, M. Rohrbach et al., "Long-term recurrent convolutional networks for visual recognition and Description," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 677–691, IEEE, 2017.

[36] Q. He, "Video content recognition technology research based on deep learning," Master Thesis, University of Electronic Science and Technology of China, Xian, China, 2017.

[37] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," 2014, https://arxiv.org/abs/1406.2199.

[38] X. Yu, Z. Zhang, L. Wu et al., "Deep ensemble learning for human action recognition in still images," *Complexity*, vol. 2020, Article ID 9428612, 23 pages, 2020.