applied sciences

MDPI

Article

Action Recognition Algorithm of Spatio–Temporal Differential LSTM Based on Feature Enhancement

Kai Hu [1,2,*], Fei Zheng [1,3], Liguo Weng [1,2], Yiwu Ding [1] and Junlan Jin [1]

1 School of Automation, Nanjing University of Information Science & Technology, Nanjing 210044, China; zhengfei@nuist.edu.cn (F.Z.); 002311@nuist.edu.cn (L.W.); dyw_0909@nuist.edu.cn (Y.D.); jjl0610@nuist.edu.cn (J.J.)
2 Jiangsu Provincial Collaborative Innovation Center for Atmospheric Environment and Equipment Technology, Nanjing University of Information Science & Technology, Nanjing 210044, China
3 China Telecom Ningbo Branch, Ningbo 315000, China
* Correspondence: 001600@nuist.edu.cn; Tel.: +86-137-7056-9871

Abstract: The Long Short-Term Memory (LSTM) network is a classic action recognition method because of its ability to extract time information. Researchers proposed many hybrid algorithms based on LSTM for human action recognition. In this paper, an improved Spatio–Temporal Differential Long Short-Term Memory (ST-D LSTM) network is proposed, an enhanced input differential feature module and a spatial memory state differential module are added to the network. Furthermore, a transmission mode of ST-D LSTM is proposed; this mode enables ST-D LSTM units to transmit the spatial memory state horizontally. Finally, these improvements are added into classical Long-term Recurrent Convolutional Networks (LRCN) to test the new network's performance. Experimental results show that ST-D LSTM can effectively improve the accuracy of LRCN.

Keywords: action recognition; Long Short-Term Memory; spatio–temporal differential

Citation: Hu, K.; Zheng, F.; Weng, L.; Ding, Y.; Jin, J. Action Recognition Algorithm of Spatio–Temporal Differential LSTM Based on Feature Enhancement. *Appl. Sci.* **2021**, *11*, 7876. https://doi.org/10.3390/app11177876

Academic Editor: Hyo Jong Lee

Received: 6 August 2021
Accepted: 24 August 2021
Published: 26 August 2021

1. Introduction

Human action recognition involves many fields, such as computer vision, image processing, deep learning, etc. It is widely used in human–computer interaction [1], video surveillance [2], intelligent transportation, sports analysis, smart home, etc. It has both academic significance and practical value. Human action recognition aims to identify action categories of moving objects and predict further actions. Its research methods are divided into two categories: one is based on manual feature extraction [3–7], and the other is based on deep learning.

The manual feature extraction method uses a traditional machine learning model to extract features from the video, then it encodes the features, standardizes the encoding vectors, trains the model, and finally carries out prediction and classification. Its advantage lies in its need-based feature extraction, strong pertinence, and simple implementation. There are noises [8] in the datasets, such as illumination, similar actions (like jogging and running), dynamic backgrounds, etc. These noises make manually extracted features ineffective in classification, so its related research is limited. Improved Dense Trajectories [9] (iDT) algorithm is one of the best algorithms based on traditional methods, and its stability is high. Many researchers combined iDT with deep learning methods to achieve higher recognition accuracy. However, the calculation speed of the iDT algorithm is very slow and it can not meet real-time requirements.

Most existing deep learning methods for action recognition are developed from convolutional neural networks. Compared with a single image, the video, which is the target of action recognition, has time-series information. Therefore, the action recognition algorithm based on deep learning pays more attention to time-series features.

In deep networks [10,11], LSTM is often applied in action recognition. It is a kind of time recurrent neural network, which is specially designed to solve the long-term

dependence problem of a general Recurrent Neural Network (RNN). Ng et al. [12] proposed a two-stream convolutional network model combined with LSTM, which can reduce computational cost and learn global video features. The two-stream convolutional network uses the CNN network (AlexNet or GoogLeNet) on ImageNet to extract image features and optical flow features of the video frames. Although the accuracy achieved by this network is only fair, it provides a new idea for the research of action recognition. Even if there is a lot of noise in optical flow images, the network combined with LSTM is helpful in classification. Du et al. [13] 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. [14] 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. Wang et al. [15] put forward the I3D-LSTM model by combining Inflated 3D ConvNets (I3D) and LSTM network; it can learn low-level and high-level features well. He et al. [16] proposed the DB-LSTM (Densely-connected Bi-directional LSTM) model; it uses dense hopping connections of Bi-LSTM (Bi-directional Long Short-Term Memory) to strengthen the feature propagation and reduce the number of parameters. This network is also an extended form of the two-stream network. Song et al. [17] used skeleton information to train the LSTM, and divided the network into two sub-networks: a temporal attention sub-network and a spatial attention sub-network.

In general, the deep learning networks of action recognition are mainly based on three types: the two-stream convolutional network, 3D convolutional network, and the LSTM network. Because the data in many practical application scenarios are generated in non-Euclidean space, the deep learning algorithm [18] meets great challenges in graph data, because the data in many practical scenarios are generated in non-Euclidean space. Therefore, action recognition algorithms based on the graph convolutional network are born. With the birth of skeletal datasets such as NTU RGB+D, action recognition algorithms based on the graph convolutional network are further developed. Most of the existing research on deep learning action recognition is based on the basic LSTM model, and many hybrid models are derived.

An action provides information in both the time domain and the space domain, and hence there are time change characteristics and space change characteristics. Although LSTM can deal with time-series information very well, it cannot deal with spatial features and features of temporal and spatial change. To make up for this shortcoming, researchers mostly increase the extraction and processing of spatial features by integrating other deep learning modules. Wang et al. [19] proposed a Spatio–Temporal LSTM (ST-LSTM) for spatio–temporal sequence prediction, which can extract spatio–temporal information. This paper further studies the ST-LSTM structure and considers its internal structure from the point of view of control theory: the ST-LSTM unit has proportional (P) and integral (I) links in the convolutional calculation and forgets temporal and spatial memory states. Compared with the typical PID control architecture, the ST-LSTM lacks the differential (D) unit. From the point of view of practical programming, the weights of gated units are always positive, and the differential calculation cannot be generated inside units. Therefore, this paper introduces the corresponding differential calculation and improves its stacked mode, to improve the feature processing on both time and space at the same time. From the point of view of robot control, the first-order differential in time represents the action speed information, and the first-order differential in space represents the position change information. The contributions of this paper are as follows:

(1) Feature enhancement is carried out. A spatio–temporal differential LSTM unit is proposed, which combines the concept of differential control in PID into the deep learning network. This modification not only considers the influence of time series and spatial position relationship on action recognition, but also increases the influence

of action speed and position change. For ST-LSTM units, a differential part is added for the temporal memory state and spatial memory state. A new LSTM unit named ST-D LSTM is designed.

(2) Feature enhancement is carried out. Due to differential calculation in ST-D LSTM units, the transfer of the two spatial states across time steps is required. Therefore, this paper designs a stacking method, that is, the horizontal transmission of spatial memory states is added. In this paper, the accuracy and stability of the stacked ST-D LSTM units are tested on different datasets; the influence of the number of stacked layers on the accuracy is studied by comparisons with other behavior recognition algorithms.

This paper is divided into five sections. Section 1 introduces the development of action recognition research. Section 2 introduces the methodology of ST-D LSTM. Section 3 introduces the ST-D LSTM unit model. Section 4 tests the performance of the ST-D LSTM model. Section 5 summarizes the work of this paper.

2. Methodology

PID control is the abbreviation of proportional integral and differential control; it has good robustness and high reliability. In the control system, the PID controller calculates the control error according to the given value and the actual output value, and then carries on proportional, integral, and differential operations on the error; finally, it combines the three operation results to obtain the control signal. Generally speaking, PID control is a linear control algorithm based on the estimation of error "past", "present", and "future" information.

Conventional PID control has three correction links: proportional, integral and differential. Their specific functions are as follows: the proportional link reflects control error proportionally, and controls the "present" error of the system. The integral controller produces the control effect at the fastest speed. It reflects the rapidity of PID control. The integral link can memorize error. In view of the "past" error of the system, the integral controller is mainly to eliminate the steady-state error. The strength of the integral function mainly depends on the integral time constant Ti. The larger Ti, the weaker the integral action. The integral function decides the accuracy of the PID control. The differential link can reflect the trend of the error (change rate). Aiming at the "future" error of the system, the differential controller improves the dynamic characteristics of the closed-loop system by acting in advance, which reflects the stability of the PID control.

After the analysis of the classic LSTM model, it is found that the recurrent memory network retains the results of the previous video frame h_{t-1} and inputs the information of current video frame x_t. The network uses different weights w_f and w_i to express the relationship between them. Moreover, it is found that when w_f and w_i are positive, it is a kind of integral (I) relation; when w_f and w_i are negative, it is a kind of differential (D) relation. Due to the weight added to video frames, this is also a proportional (P) relationship. When referring to the code of the ST-LSTM on the Github, it is found that w_f and w_i are positive, so for the ST-LSTM, its internal temporal memory state and spatial memory state have a proportional (P) and integral (I) relationship. From the point of view of PID control, the differential link in the ST-LSTM is missing, so we try to add a differential (D) to the ST-LSTM. From the perspective of deep learning, adding differential is also an idea of feature enhancement.

From the point of view of robot kinematics, action characteristics include posture, position, speed, etc. Taking the manipulator of a robot as an example, the action of the arm includes the translation of the center of mass and the rotation around the centroid. When the manipulator is analyzed by the Newton–Euler equation, the dynamic equation is as follows:

$$\tau = M(\theta)\ddot{\theta} + V(\theta, \dot{\theta}) + G(\theta) \tag{1}$$

In the above formula, $M(\theta)$ is the $n \times n$ mass matrix of the operating arm, $V(\theta, \dot{\theta})$ is the centrifugal force and the Gordian force vector of $n \times 1$. $G(\theta)$ is the gravity vector of $n \times 1$, which depends on the position and velocity. $M(\theta)$ and $G(\theta)$ are complex functions

about positions of all joints of the operating arm θ. $\dot{\theta}$ represents the angle velocity. $\ddot{\theta}$ represents the acceleration. Therefore, in the control theory, the control of the robot needs a differential state.

The action recognition network based on deep learning pays attention to the extraction of action posture information. Enhancing the information extraction of limb speed and position changes can improve the final performance of the network. Velocity and position changes are the first-order differential of action temporal state and spatial state, respectively. Therefore, the differential of PID control is introduced into the ST-LSTM to extract more information such as gesture and velocity position changes.

Moreover, although the ST-LSTM increases the influence of the spatial series on the gesture, the time series taken into account by a unit is only the current time series and the last time series. Due to the proportional relationship in the forgetting gate, only part of the previous time series is retained. However, for a complete action, the action is continuous, a complete action cannot be completed in only two short time series. A simple action (such as bowing) needs at least 3–4 time series to complete, and there are actions which are more complex and need more time series to complete. Therefore, it is necessary to retain more time-series information.

Based on the above ideas, the Spatio–Temporal Differential LSTM unit is proposed, it combins the ST-LSTM with a differential module. Moreover, a basic and a multi-layer LSTM are built, to show the performance of the improved differential LSTM network. It is shown that the ST-D LSTM can improve the recognition performance and can capture more action information. The ST-D LSTM can be flexibly embedded into different networks to achieve different applications.

This paper uses the idea of differential control in PID control. The input differential can capture the speed information, and the temporal state differential can capture the change information of action position. The improved ST-D LSTM unit can improve the accuracy of action recognition, and increase the stability of the network.

3. ST-D LSTM

Although researchers made some progress in accuracy, the framework of most algorithms is too complex. The improvement of accuracy depends on the network depth and the number of parameters. This paper proposes the ST-D LSTM structure based on spatio–temporal differential and the suitable stacking method. In order to better demonstrate its performance and usage, we used ST-D LSTM to replace LSTM in the classic LRCN. The network structure can simultaneously take into account temporal and spatial information and complete the transmission of spatial information changes across time steps. In the process of information transmission, the horizontal structure pays attention to the feature extraction on the time flow, and the vertical structure pays attention to the feature extraction on the spatial flow. Moreover, the input differential increases the feature extraction of the limb movement speed. The spatial differential information across video frames can increase the feature extraction of the position changes in different frames. The combination of horizontal and vertical information transmission mode enables the network to combine temporal and spatial features and corresponding features, to make the final judgment. This method can extract more action features without adding other deep learning modules, achieve better recognition accuracy and avoid increasing the network complexity.

3.1. The Internal Structure of the ST-D LSTM

Wang et al. [19] proposed the ST-LSTM structure for spatio–temporal sequence prediction; it can realize information transmission between different layers of LSTM units.

ST-LSTM is improved based on the ConvLSTM [20] structure. Vertically, spatial information memory states between the LSTM units at different layers are similar to the horizontal memory states of the ConvLSTM unit, and the spatio–temporal memory module is added based on the original horizontal memory state. The ST-LSTM transmits the information of hidden layers, and increases the transmission of spatial information in the

vertical direction, to realize the transmission of memory information between different layers in this time step. ST-LSTM is the core part of the PredRNN algorithm.

For action recognition, limb position change is a vital feature; that is, the time change and position change should be considered at the same time. The zigzag transfer method enables the stacked ST-LSTM unit to transfer the spatial state longitudinally at each time step. Although the PredRNN algorithm considers both temporal and spatial features through the zigzag cross-layer connection, it ignores changes of temporal and spatial features. For this reason, the SpatioTemporal Differential LSTM (ST-D LSTM) unit is proposed, with the idea of spatio–temporal variation based on the spatial memory state of the ST-LSTM unit.

The ST-D LSTM is similar to the LSTM. It also contains the forgetting gate, the input gate, and the output gate. Furthermore, the ST-D LSTM unit also contains two cell states: the temporal memory module C_{t-1}^{l} and the spatial memory module S_{t}^{l-1}. The temporal memory module stores the temporal characteristic information of the previous $t-1$ moments in the same layer units, while the spatial memory module stores the spatial characteristic information of different layer units. x_t represents input in the ST-D LSTM unit; h_{t-1}^{l} is the hidden layer state. k_t, i_t and f_t are the conversion mechanism, the input gate and the output door of temporal memory, respectively. k'_t, i'_t and f'_t are the conversion mechanism, the input, and the output door of the spatial memory, respectively. The output gate o_t combines temporal memory and spatial memory.

Similarly to the differential part in the PID control, the differential module of spatial memory state is added to the original LSTM unit according to the connection mode of the input gate. The "future" error, that is the characteristic change information, is introduced into the present state by integral calculation, so that the network can improve the accuracy and stability. In addition, the input differential module is added at the same time to increase the propagation of spatial features in the same layer of the LSTM unit along the horizontal time step, so that the network can take into account the temporal information, the limb moving speed and trajectory. The ST-D LSTM internal structure diagram is shown in Figure 1.

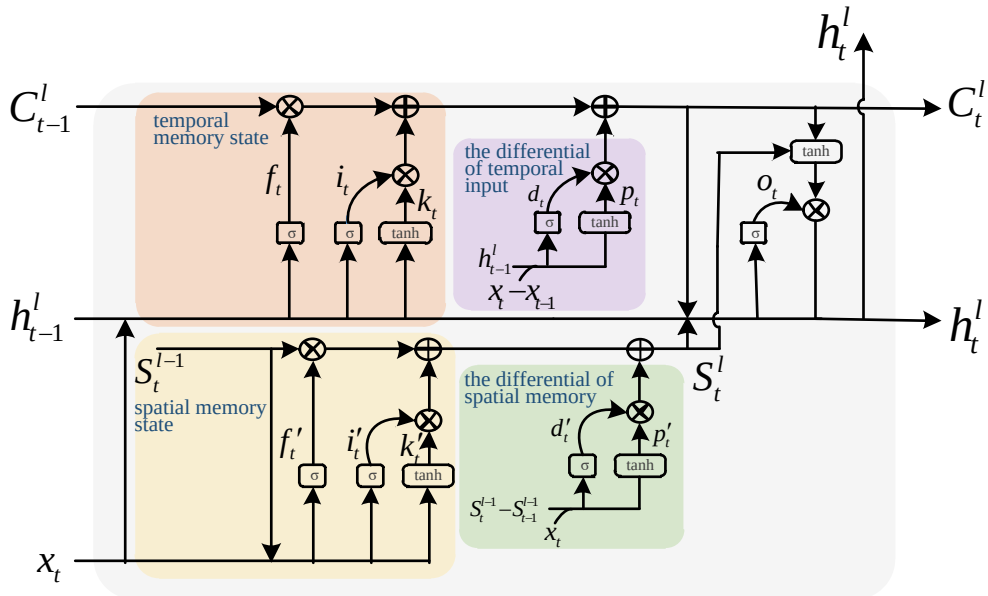


Figure 1. The internal structure diagram of the ST-D LSTM.

In the mathematical model, t is a small value, so the input differential $\frac{\mathrm{d}x(t)}{dt}$ is approximated to $x_t - x_{t-1}$, that is $\frac{\mathrm{d}x(t)}{dt} \approx x_t - x_{t-1}$. Similarly, the spatial memory differential can be expressed as $S_t^{l-1} - S_{t-1}^{l-2}$. Approximation can make the calculation easier while realizing the differentiation of the input and spatial state. The differential processing is

similar to the optical flow method in image processing. The input differentiation provides information on the image speed change and the spatial memory differentiation provides the position change information of the image.

In this paper, the LRCN network framework is used for subsequent experiments, and the input to the ST-D LSTM unit is features extracted by the CNN, so convolutions are not used in the ST-D LSTM unit, and each gate can be considered a fully connected connection. The temporal memory state equations of the forgetting gate, input gate, and input differentiation in the ST-D LSTM unit are shown in Equations (2) and (3):

$$\begin{pmatrix} f_t \\ i_t \\ k_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \tanh \end{pmatrix} \left(W \cdot \left[x_t, h^l_{t-1} \right] \right) \tag{2}$$

$$\begin{pmatrix} d_t \\ p_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \tanh \end{pmatrix} \left(W \cdot \left[x_t - x_{t-1}, h^l_{t-1} \right] \right) \tag{3}$$

The spatial memory equations of the forgetting gate, input gate and differentiation in the ST-D LSTM unit are shown in Equations (4) and (5):

$$\begin{pmatrix} f'_t \\ i'_t \\ k'_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \tanh \end{pmatrix} \left(W \cdot \left[x_t, S^{l-1}_t \right] \right) \tag{4}$$

$$\begin{pmatrix} d'_t \\ p'_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \tanh \end{pmatrix} \left(W \cdot \left[x_t, S^{l-1}_t - S^{l-1}_{t-1} \right] \right) \tag{5}$$

When $l = 1$, $S^{l-1}_t = S^L_t, S^{l-1}_{t-1} = S^L_{t-2}$.

The updated temporal cell state and spatial cell state are:

$$C^l_t = f_t \circ C^l_{t-1} + i_t \circ k_t + d_t \circ p_t \tag{6}$$

$$S^l_t = f'_t \circ S^{l-1}_t + i'_t \circ k'_t + d'_t \circ p'_t \tag{7}$$

The equation of the output gate in the ST-D LSTM unit is:

$$O_t = \sigma(w_O \cdot [h^l_{t-1}, C^l_t, S^l_t, x_t] + b_O) \tag{8}$$

$$\mathrm{h}^l_t = O_t \circ \tan(C^l_t, S^l_t) \tag{9}$$

3.2. The Stacked Mode of the ST-D LSTM Unit

The differential calculation of spatial states in ST-D LSTM units requires the transmission of spatial memory in the same layer across two steps. To cooperate with the spatial state differentiation, an improved transfer method of state memories is proposed. The spatial memory at each step is divided into horizontal and vertical transmission after output, and the differential calculation is carried out outside the unit. This method will not increase the amount of data in transmission, so the speed of the network will not be too slow. The connection is shown in Figure 2.

As shown in Figure 2, based on the traditional LSTM cell stacked mode, and with reference to the vertical propagation of the PredRNN spatial memory state, the split propagation is carried out to increase the horizontal transmission of the spatial memory. Moreover, the differential calculation is carried out outside the unit; that is, the differentiation between the spatial memory of the upper layer at this time step S^{l-1}_t and S^{l-1}_{t-1} the spatial memory at the previous step is added. In this connection mode, the temporal memory state is only transmitted horizontally, and the temporal information features extracted by each layer are partially retained and input to the next layer. The horizontal transmission of the spatial memory state makes the location feature changes with the same precision rate to be

transmitted. For the unit in the first layer at time t, the differentiation between the spatial memory state of the previous time step S^l_{t-1} and that of the time step S^l_{t-2} is added; that is, $S^l_{t-1} - S^l_{t-2}$. The spatial memory state output of the unit is divided into two directions, one direction continues the longitudinal spatial memory transmission, and the other direction performs the differential calculation. This connection mode can increase the information of position change without affecting the calculation speed, and subsequent experiments will verify its effectiveness.

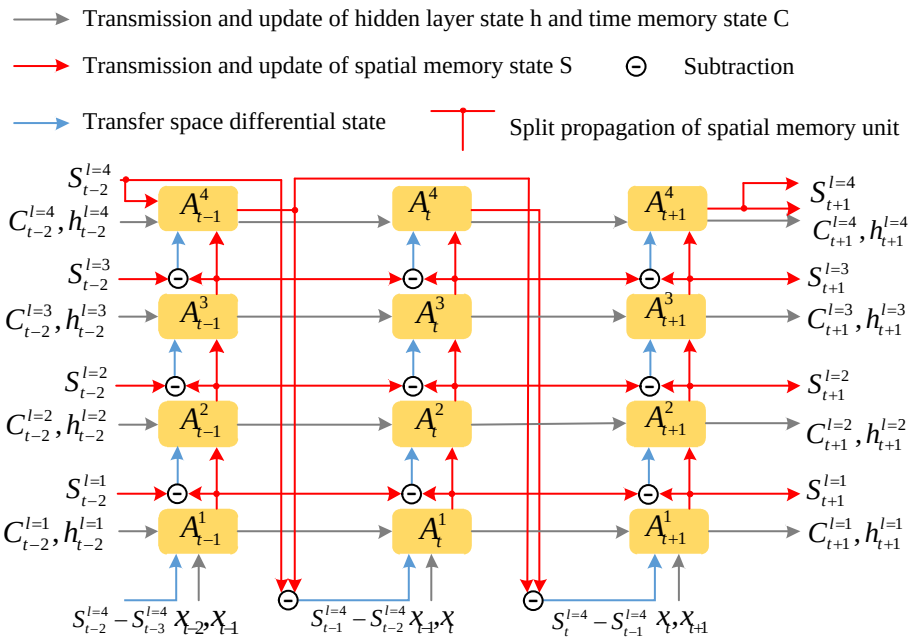


Figure 2. The connection mode between ST-D LSTM units.

4. Experiments

In order to show the performance of the ST-D LSTM unit, this section carries out experiments on the three datasets, UCF-101, HMDB-51, and Hollywood2. The results directly prove its advantages in accuracy, and the influence of the stack number of ST-D LSTM units on recognition accuracy is further studied. Finally, this section compares the recognition accuracy of the ST-D LSTM unit with other algorithms on UCF-101 and HMDB-51.

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 essential functions:

(1) The researchers do not have to consider the process of collection and pretreatment.
(2) It is able to compare different algorithms under the same standard.

The KTH dataset [21] was released in 2004. The KTH dataset includes six kinds of actions (including strolling, 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 similar.

The Weizmann dataset [22] was released in 2005 and includes nine people completing ten kinds of actions (bending, stretching, high jump, jumping, running, standing, hopping, walking, waving1, and waving). 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 are released early. The citation rate of these datasets is high. However, with the rapid development of action recognition, there are shortcomings: the

background is simple, the angle is fixed, and each video has only one person. The above two datasets already cannot satisfy actual action recognition requirements, so they are rarely used now.

The Hollywood2 dataset [23] was released in 2009. The video data in the dataset are collected from Hollywood movies. There are 3669 video clips in total, including 12 action categories (such as: 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 [24] in 2012. The dataset samples include various action samples collected from TV stations and video samples saved from 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.

Brown University released the HMDB-51 dataset [25] in 2011. The samples come from video clips of 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 types and a wide range of actions. The scenes in the Hollywood2 dataset are more complex and closer to real life. To comprehensively verify the ST-D LSTM unit's performance, three datasets, UCF-10, HMDB-51, and Hollywood2, were chosen for training and testing. Furthermore, the ST-D LSTM unit's performance was tested in the above three databases, respectively. The UCF-101 and HMDB-51 datasets are commonly used in deep learning algorithms, so these two datasets were used when the ST-D LSTM unit was compared with other deep learning-based algorithms.

4.2. Method

To test the accuracy of the ST-D LSTM, a simple Long-term Recurrent Convolutional Network [26] (LRCN) is adopted in experiments.

The LRCN connects the stacked LSTM model directly with the CNN; it extracts the spatial features of the pre-trained CNN and inputs spatial features to the LSTM model to learn the temporal and spatial features at the same time. The framework of LRCN is shown in Figure 3. The model first converts the video to frame images, then uses the pre-trained CNN to extract the spatial features of the frame images; next, it inputs the extracted features into the ST-D LSTM network to extract the temporal and spatial information. As a result, the network learns the temporal relationship from spatial features of frame images. Finally, the result is classified by Softmax.

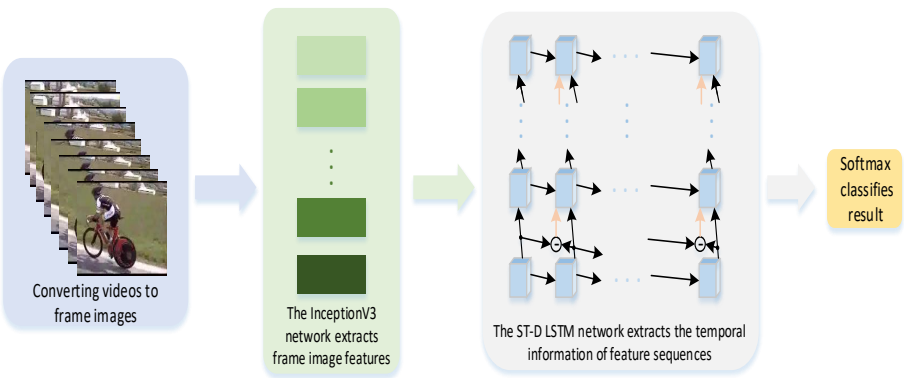


Figure 3. The LRCN network framework based on the ST-D LSTM.

In the experiment, the convolutional network is used to extract spatial features and the LSTM network is used to extract temporal features. However, it is slightly different from the original LRCN. In CNN feature extraction, the InceptionV3 with less computation but high performance is used to extract image features. In the LSTM network, the number of

hidden layers is defined according to the requirements of computer performance, and the LSTM unit uses the ST-D LSTM unit.

The ST-D LSTM unit is applied to the network model in Figure 3, and is evaluated in terms of accuracy, loss and standard deviation. To better show the improved LSTM units' performance, 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 are consistent. The batch_size is 32, the number of the hidden layers is 5, the hidden layers' parameter is 1024, the full connection layers' parameter is 512, and the loss function is the classic cross-entropy function. In the follow-up experiments, one layer, two layers, three layers, four layers, and five hidden layers are used to study the influence of the number of hidden layers on recognition accuracy.

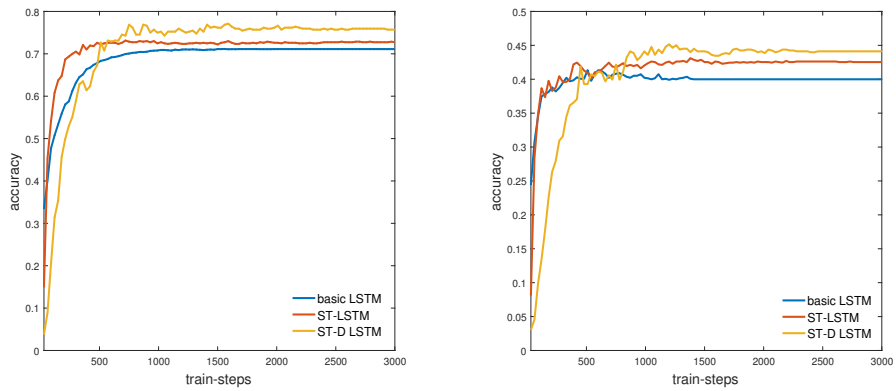
The assessment method is the direct hold-out method. To avoid the data division influencing the result and increase the final evaluation result's fidelity, the training set and the testing set are divided in the same way at each type of action in every dataset in the experiment. The training set accounts 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 paper uses multiple hold-outs to take the average of the results. Each LSTM unit uses the hold-out method to divide the dataset. After an experiment is concluded, the dataset is re-divided, and the experiment is performed again, and this is then repeated. The experiments were performed using three datasets of five different LSTM units, each repeated three times. At last, the average accuracy of three experimental results is the result of the LSTM unit.

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

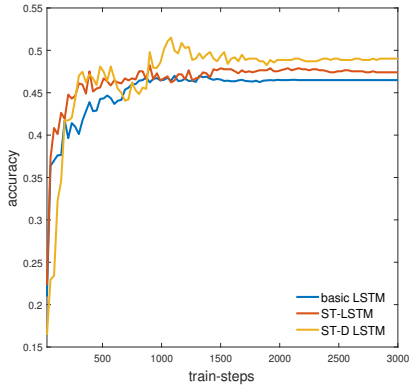
4.3.1. The Influence of Internal Structure on Accuracy

In this experiment, the LRCN network was selected as the basic network framework. The basic LSTM unit, ST-LSTM unit and ST-D LSTM unit were used in the stacking part of the LSTM, and the common connection mode; the zigzag connection mode and the differential connection mode corresponding to each unit were selected. The number of hidden layers was 5 and the parameter was set to 1024. Figures 4 and 5 show the comparison of accuracy and loss optimization of the basic LSTM unit, ST-LSTM unit, and ST-D LSTM unit in the three datasets, respectively.



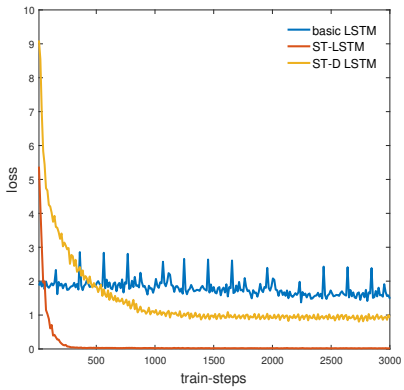
(**a**) UCF-101 (**b**) HMDB-51

Figure 4. *Cont.*

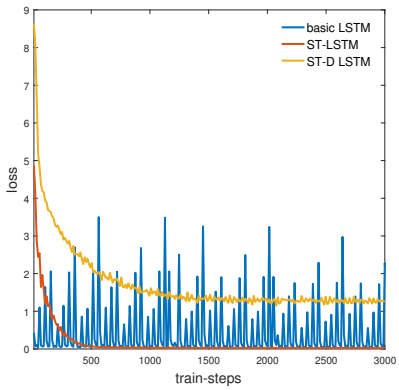


(**c**) Hollywood2

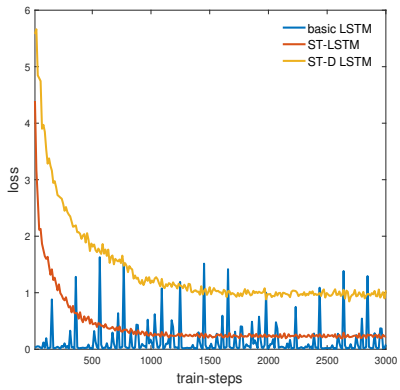
Figure 4. The comparison of different LSTM units on three datasets in accuracy.



(**a**) UCF-101



(**b**) HMDB-51



(**c**) Hollywood2

Figure 5. The comparison of different LSTM units on three datasets in loss.

Figure 4 shows the accuracy of the basic LSTM unit, ST-LSTM unit, and ST-D LSTM unit. Table 1 shows the final accuracy when the accuracy reaches a stable stage. As shown in Figure 4 and Table 1, due to the differential transmission, the accuracy of the ST-D LSTM unit is the slowest to reach the stable stage, but its final recognition accuracy is the highest. Thus, the temporal state differential and input differential modules can increase the extraction and improve the accuracy.

As shown in Figure 5, the loss of the ST-D LSTM can finally converge to a stable stage, but the convergence rate and the final convergence value are slightly lower than those of the ST-LSTM, which may be caused by the differential module. To objectively compare the loss optimization processes, the same loss function and optimizer are used in different

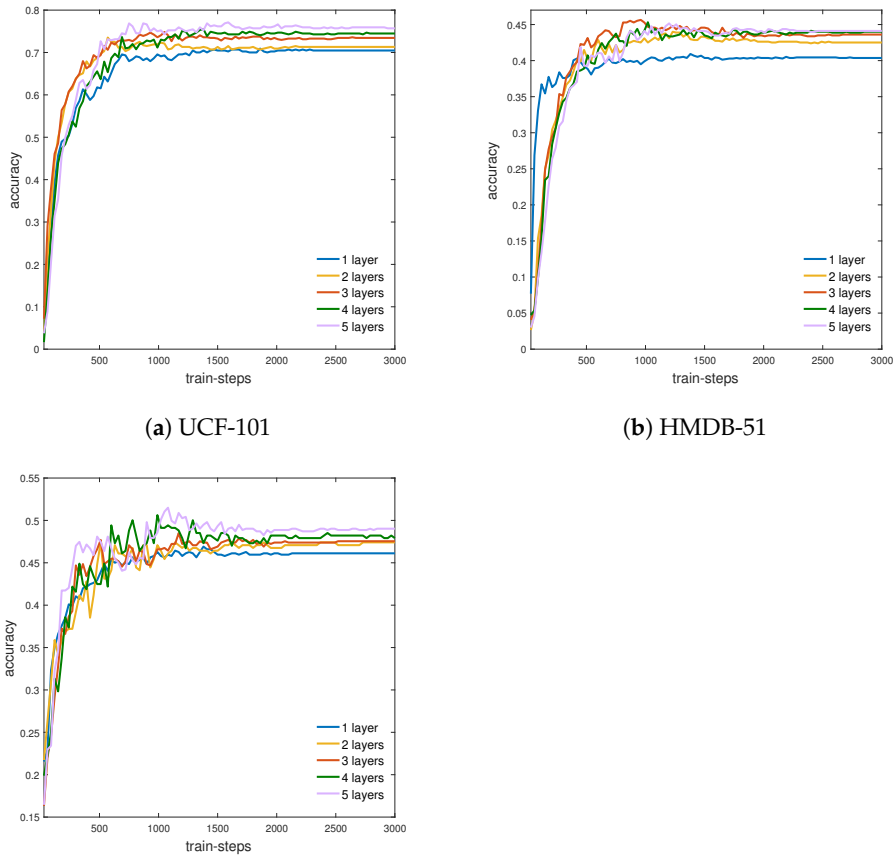
LSTM units. It can be found that the loss value of ST-D LSTM unit still has room to be optimized, and the loss function can be further designed and optimized.

Table 1. The accuracy of different LSTM units on three datasets.

	UCF-101	HMDB-51	Hollywood2
basic LSTM	71.15%	39.99%	46.49%
ST-LSTM	72.73%	42.53%	47.41%
ST-D LSTM	75.70%	44.11%	49.02%

4.3.2. The Influence of the Number of Stacking Layers

In the performance verification and comparison experiment, the recognition accuracy obtained by stacking five-layer ST-D LSTM units was used. However, in the actual process of parameter adjustment, it can be found that the performance of stacking different layers of ST-D LSTM units is different in accuracy and training speed. Therefore, the ST-D LSTM units are stacked one layer, two layers, three layers, four layers, and five layers, respectively, and the LRCN network is applied for experiments. In this experiment, only the number of layers varies, the other parameters such as batch size, parameters of the hidden layer, training steps and so on are consistent. The process of accuracy climbing is shown in Figure 6, and the stable accuracy is shown in Table 2.



(**a**) UCF-101

(**b**) HMDB-51

(**c**) Hollywood2

Figure 6. The comparison of the accuracy increasing process of stacked ST-D LSTM units with different layers.

When ST-D LSTM units with different layers are stacked, there is a significant difference in training speed. The impacts are studied from two aspects of accuracy and training speed. The network training speed is shown in Table 3. In the speed experiment, the fps index is used, that is, the number of video frames processed in one second.

Table 2. The accuracy comparison of stacked ST-D LSTM units with different layers.

	UCF-101	HMDB-51	Hollywood2
1 layer	70.47%	40.39%	46.12%
2 layers	71.32%	42.51%	47.41%
3 layers	73.44%	43.61%	47.54%
4 layers	74.48%	44.01%	48.21%
5 layers	75.70%	44.11%	49.02%

Table 3. The training speed comparison of stacked ST-D LSTM units with different layers (in frames per second).

	UCF-101	HMDB-51	Hollywood2
1 layer	38	35	56
2 layers	31	24	42
3 layers	27	17	34
4 layers	16	10	23
5 layers	11	10	14

Through experiments, it can be found that increasing the number of layers can improve the accuracy. When five layers are stacked, ST-D LSTM units perform the best on the HMDB-51, UCF-101, and Hollywood2 datasets. However, increasing layers will also increase the time needed for reading data and training. Stacking too many layers will slow down the training. When studying the translation task based on LSTM, Wu et al. [27] found that the network can work well by simply stacking four layers of LSTM units, and six layers is the limit. Stacking more than eight layers makes the network fail. Table 2 shows that when ST-D LSTM units are stacked to layers 4 and 5 on the HMDB-51 dataset, the recognition accuracy only increases slightly. Therefore, although stacked LSTM layers can increase network performance, in general, the LSTM units can better balance the training speed and accuracy with 4–5 stacked layers.

4.3.3. Comparison of ST-LSTM and ST-D LSTM in Terms of Stability and Accuracy

For stability experiments, the ST-LSTM and ST-D LSTM units, which are both stacked five-layers, were applied to the LRCN network for three repeated experiments. The average accuracy was calculated as the final result. The standard deviation was calculated to compare the stability of the ST-LSTM unit and ST-D LSTM unit. The average accuracy and standard deviation of the three repeated experiments are plotted. As shown in Figure 7, in three different datasets, the accuracy of the ST-D LSTM unit is higher than that of the ST-LSTM unit, but the standard deviation is not higher than that of the ST-LSTM unit. Therefore, the ST-D LSTM unit has good stability.

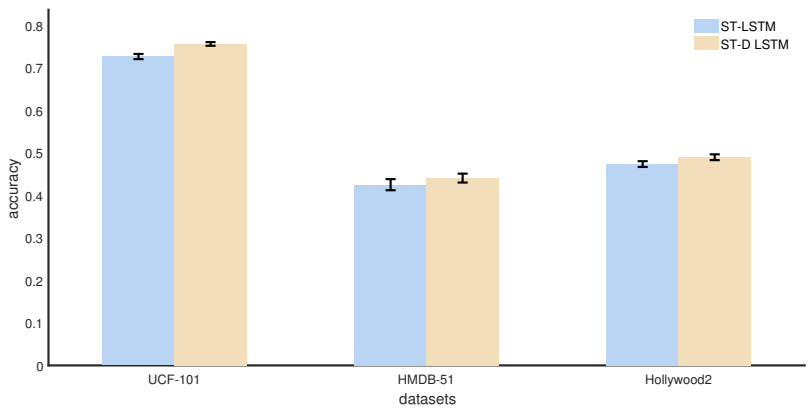


Figure 7. The comparison of accuracy and standard deviation between the ST-LSTM and ST-D LSTM.

In order to further verify the performance of the ST-D LSTM unit, the ST-D LSTM unit is compared with other deep learning algorithms. The experiment is performed on the UCF-101 and HMDB-51 datasets and results are shown in Table 4.

Table 4. The accuracy comparison of various deep learning algorithms on UCF-101 and HMDB-51 datasets.

		UCF-101	HMDB-51
Two-stream Convolutional Network [28]		73.00%	40.50%
LRCN	basic LSTM	71.15%	39.99%
	ST-LSTM	72.73%	42.53%
	BiLSTM [22]	70.00%	39.81%
	LSTM+attention	72.40%	41.50%
	ST-D LSTM	75.70%	44.11%

The ST-D LSTM is compared with the two-stream convolutional network, the LRCN network with an attention mechanism, and the LRCN network with BiLSTM. Due to differential calculation, the ST-D LSTM unit is more sensitive to action changes and can achieve high accuracy on the UCF-101 and the HMDB-51 datasets.

5. Conclusions and Prospect

Human action recognition has many applications in today's society. Although existing networks can achieve good accuracy, many have limitations in application scenarios. In this paper, the internal structure of the LSTM unit is improved. A ST-D LSTM unit with high accuracy and high reliability is proposed and applied to action recognition. The ST-D LSTM unit updates and transmits action spatial feature change information: the differential operation of the spatial memory state is carried out in the process of transmission, and hence the ST-D LSTM has proportional, integral and differential operations. The ST-D LSTM can satisfy the requirements of rapidity, accuracy, and stability. In the verification experiments, the accuracy of the ST-D LSTM unit is better than that of the ST-LSTM unit in the UCF-101, HMDB-51, and Hollywood2 datasets, and its stability is no less than that of the ST-LSTM unit. However, due to the methods of data reading and transferring in deep learning, the differential calculation leads to a double increase in the amount of data. Therefore, the speed of the ST-D LSTM network cannot be guaranteed, and the amount of parameters needs to be further optimized. Compared with other action recognition algorithms based on deep learning, the ST-D LSTM unit shows good accuracy in the UCF-101 and HMDB-51 datasets. The ST-D LSTM unit is applied to the LRCN network in the experiments. Because the LRCN algorithm extracts features before processing them, the LRCN network applying in the ST-D LSTM unit does not achieve the end-to-end training. In the follow-up research, the ST-D LSTM unit can use convolutional calculations in the internal structure. The ST-D LSTM unit can be applied to other network frameworks to achieve the end-to-end training. Moreover, the ST-D LSTM unit can also be applied to other scenarios, such as attitude estimation, sequence prediction, and so on.

Author Contributions: Conceptualization, K.H. and F.Z.; methodology, K.H.; software, F.Z.; validation, F.Z., Y.D. and J.J.; formal analysis, F.Z., J.J.; investigation, L.W.; resources, K.H.; data curation, K.H.; writing—original draft preparation, F.Z.; writing—review and editing, F.Z., L.W. and K.H; visualization, F.Z.; supervision, K.H.; project administration, K.H.; funding acquisition, K.H. All authors have read and agreed to the published version of the manuscript

Funding: The research in this paper is supported by the National Natural Science Foundation of China (42075130), Industry prospect and key core technology key projects of Jiangsu Province (BE2020006-2), the key special project of the National Key R&D Program (2018YFC1405703), NUIST Students' Platform for Innovation and Entrepreneurship Training Program (202010300050Z). I would like to express my heartfelt thanks to the reviewers who submitted valuable revisions to this article.

Institutional Review Board Statement: Ethical review and approval were waived for this study, due to the data being provided publicly.

Informed Consent Statement: Not applicable.

Data Availability Statement: 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 (https://serre-lab.clps.brown.edu/resource/hmdb-a-large-human-motion-database/, accessed on 23 August 2021), UCF-101 (www.crcv.ucf.edu/data/UCF101.php, accessed on 23 August 2021), Hollywood2 (www.di.ens.fr/~laptev/actions/hollywood2/, accessed on 23 August 2021).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Roitberg, A.; Perzylo, A.; Somani, N.; Giuliani, M.; Rickert, M.; Knoll, A. Human activity recognition in the context of industrial human-robot interaction. In Proceedings of the Signal and Information Processing Association Annual Summit and Conference (APSIPA), Chiang Mai, Thailand, 9–12 December 2014; pp. 1–10.
2. Wang, H.; Kläser, A.; Schmid, C.; Liu, C.L. Dense trajectories and motion boundary descriptors for action recognition. *Int. J. Comput. Vis.* **2013**, *103*, 60–79. [CrossRef]
3. Yang, X.; Tian, Y.L. Action Recognition Using Super Sparse Coding Vector with Spatio-temporal Awareness. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 727–741.
4. Peng, X.; Zou, C.; Qiao, Y.; Peng, Q. Action Recognition with Stacked Fisher Vectors. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 581–595.
5. Peng, X.; Wang, L.; Wang, X.; Yu, Q. Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice. *Comput. Vis. Image Underst.* **2016**, *150*, 109–125. [CrossRef]
6. Arandjelovic, R.; Zisserman, A. All about VLAD. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 1578–1585.
7. Duta, I.C.; Ionescu, B.; Aizawa, K.; Sebe, N. Spatio-temporal vlad encoding for human action recognition in videos. In Proceedings of the International Conference on Multimedia Modeling, Reykjavik, Iceland, 4–6 January 2017; pp. 365–378.
8. Zhu, H.; Zhu, C.; Xu, Z. Research advanves on human activity recognition datasets. *Acta Autom. Sin.* **2018**, *44*, 978–1004.
9. Wang, H.; Schmid, C. Action recognition with improved trajectories. In Proceedings of the IEEE International Conference on Computer Vision, Sydney, Australia, 1–8 December 2013; pp. 3551–3558.
10. Chen, B.; Xia, M.; Huang, J. Mfanet: A multi-level feature aggregation network for semantic segmentation of land cover. *Remote Sens.* **2021**, *13*, 731. [CrossRef]
11. Xia, M.; Zhang, X.; Weng, L.; Xu, Y. Multi-stage feature constraints learning for age estimation. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 2417–2428. [CrossRef]
12. Ng, Y.H.; Hausknecht, M.; Vijayanarasimhan, S.; Vinyals, O.; Toderici, G. Beyond short snippets: Deep networks for video classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 4694–4702.
13. Du, W.W.; Wang, Y.; Qiao, Y. Rpan: An end-to-end recurrent pose-attention network for action recognition in videos. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 3725–3734.
14. Long, X.; Gan, C.; De, M.G.; Liu, X.; Li, Y.; Li, F.; Wen, S. Multimodal keyless attention fusion for video classification. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
15. Wang, X.; Miao, Z.; Zhang, R.; Hao, S. I3d-lstm: A new model for human action recognition. In Proceedings of the IOP Conference Series: Materials Science and Engineering, Jakarta, Indonesia, 29–31 March 2014; p. 32035.
16. He, J.Y.; Wu, X.; Cheng, Z.Q.; Yuan, Z.; Jiang, Y.G. DB-LSTM: Densely-connected Bi-directional LSTM for human action recognition. *Neurocomputing* **2021**, *444*, 319–331. [CrossRef]
17. Song, S.; Lan, C.; Xing, J.; Zeng, W.; Liu, J. An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2014.
18. Xia, M.; Wang, T.; Zhang, Y.; Liu, J.; Xu, Y. Cloud/shadow segmentation based on global attention feature fusion residual network for remote sensing imagery. *Int. J. Remote Sens.* **2021**, *42*, 2022–2045. [CrossRef]
19. Wang, Y.; Long, M.; Wang, J.; Gao, Z.; Yu, P.S. Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 879–888.
20. Shi, X.; Chen, Z.; Wang, H.; Yeung, D.Y.; Wong, W.K.; Woo, W.C. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In Proceedings of the Advances in Neural Information Processing Systems, Palais des Congrès de Montréal, Montréal, QC, Canada, 7–10 December 2015; pp. 802–810.
21. Schulst, C.; Laptev, I.; Caputo, B. Recognizing human actions: A local SVM approach. In Proceedings of the 17th International Conference on Pattern Recognition, Cambridge, UK, 23–25 August 2004; pp. 32–36.

22. Moshe, B.; Lena, G.; Eli, S.; Michal, I.; Ronen, B. Actions as Space-Time Shapes. In Proceedings of the Tenth IEEE International Conference on Computer Vision, Beiging, China, 17–20 October 2005; pp. 1395–1402.
23. Marszalek, M.; Laptev, I.; Schmid, C. Actions in context. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 2929–2936.
24. Soomro, K.; Zamir, A.R.; Shah, M. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv* **2012**, arXiv:1212.0402.
25. Kuehne, H.; Jhuang, H.; Garrot, E.; Poggio, T.; Serre, T. HMDB: A large video database for human motion recognition. In Proceedings of the International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 2556–2563.
26. Donahue, J.; Anne, H.L.; Guadarrama, S.; Rohrbach, M.; Venugopalan, S.; Saenko, K.; Darrell, T. Long-term recurrent convolutional networks for visual recognition and description. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 2625–2634.
27. Wu, Y.; Schuster, M.; Chen, Z.; Le, Q.V.; Norouzi, M.; Macherey, W.; Macherey, W. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv* **2016**, arXiv:1609.08144.
28. Simonyan, K.; Zisserman, A. Two-stream convolutional networks for action recognition in videos. *arXiv* **2014**, arXiv:1406.2199.