Dynamic Gesture Recognition via Fusion of Wide Residual and Long Short-Term Memory Networks: Postprint

Authors: Liang Zhijie, Liao Shengbin

Date: 2018-10-11T00:00:00+00:00

Abstract

To address the issue that existing dynamic gesture recognition methods struggle to accurately match spatiotemporal features in long-term sequences, we propose a spatiotemporal feature consistent gesture recognition method based on wide residual and bidirectional long short-term memory networks. First, a pretrained 3D convolutional neural network is employed to synchronously extract short-term features from both spatial and temporal dimensions of videos, which are then synchronously parsed by a bidirectional spatial long short-term memory network to form long-term spatiotemporal feature connection units, serving as input to the residual network. To validate the effectiveness of the proposed algorithm, a novel multimodal gesture dataset was constructed using Kinect sensors. Experiments on three public gesture recognition datasets—SLVM, Montalbano, and SKIG—demonstrate that the proposed method achieves excellent performance, with recognition accuracy surpassing the currently published state-of-the-art rates.

Full Text

Preamble

Dynamic Gesture Recognition Based on Wide Residual Networks and Long Short-Term Memory Networks

Liang Zhijie[1,2], Liao Shengbin[1] [1]National Engineering Research Center for E-Learning, Central China Normal University, Wuhan 430079, China [2]School of Adult & Network Education, Southwest University of Science and Technology, Mianyang, Sichuan 621010, China

Abstract: Existing dynamic gesture recognition methods struggle to accurately match spatiotemporal features in long-duration sequences. To address

this limitation, we propose a novel spatiotemporal feature consistency approach for gesture recognition based on wide residual networks and bidirectional long short-term memory networks. Our method first employs a pretrained 3D convolutional neural network to extract short-term features synchronously from both spatial and temporal dimensions of video data. These features are then processed by a bidirectional spatial LSTM to form long-term spatiotemporal connection units, which serve as input to the residual network. To validate the effectiveness of our algorithm, we constructed a new multimodal gesture dataset using Kinect sensors. Experiments on three public gesture recognition datasets —SLVM, Montalbano, and SKIG—demonstrate that our method achieves excellent performance, surpassing the currently published state-of-the-art recognition rates.

Keywords: gesture recognition; 3D convolutional neural networks; long short-term memory networks; wide residual networks

0 Introduction

Gesture serves as the most powerful communication tool among the deaf community and represents a crucial pathway for deaf individuals to access information services and share societal and cultural achievements with the hearing population. Gestures also offer natural and intuitive visual effects, making them highly promising for human-computer interaction applications. Consequently, numerous research institutions and scholars worldwide have investigated gesture recognition algorithms to enable machines to automatically understand human gestures. However, gesture recognition remains a challenging research problem due to the complex deformable nature of the human hand, the diversity and ambiguity of gestures, and particularly the temporal distribution differences inherent in dynamic gestures.

Early explorations in gesture recognition for human-computer interaction, such as the work of Pavlovic et al. [**?**], focused on using artificial neural networks to recognize static hand shapes. As hardware systems advanced, algorithmic complexity in static gesture recognition continued to increase. For instance, Dardas et al. [**?**] employed support vector machine (SVM) algorithms to study static gesture recognition representing digits 0 through 9, achieving high recognition rates in real-time environments. Generally, static gesture recognition requires only a single image, whereas dynamic gesture recognition presents greater challenges in accuracy and usability due to its flexible, variable nature and rich vocabulary.

Most current dynamic gesture recognition methods rely on manual feature engineering. Parcheta et al. [**?**] investigated dynamic gesture recognition using hidden Markov models (HMM) on a self-created public dataset containing 91 sign language vocabulary items, achieving 84.6% accuracy. Daniel et al. [**?**] proposed an alternative sign language recognition system fusing HMM and dynamic

time warping (DTW), which utilized HMM for accurate tracking of upper limb motion trajectories and achieved 95.1% recognition accuracy on the Cambridge Gestural Performance Database 2012 (CGPD12) [**?**]. In domestic research, Cao et al. [**?**] first performed dynamic gesture contour segmentation on RGB-D images using K-means clustering, then combined fast dynamic time warping for recognition, achieving 96.8% accuracy. Zhang et al. [**?**] used Kinect sensors to obtain human joint data for building training template libraries, subsequently employing DTW algorithms for high-precision recognition of traffic police command gestures.

However, manual feature extraction and selection is extremely time-consuming and labor-intensive, requiring deep domain expertise to ensure the correctness of classification features. Moreover, manually selected features struggle to adapt to the variability of dynamic gestures. In recent years, with further improvements in computer hardware performance, the information technology field has witnessed a new wave of artificial intelligence transformation, particularly deep learning based on neural networks [**?**], which has attracted unprecedented attention. Unlike traditional approaches combining manual feature extraction with classifiers, deep learning integrates automatic feature extraction and classification into an end-to-end learning architecture, avoiding the subjectivity of manual feature engineering and achieving qualitative improvements in recognition rates.

Moon et al. [**?**] studied dynamic gesture recognition using ordinary monocular cameras as data sensors, proposing a large-scale dataset gesture recognition method based on convolutional neural networks (CNN). To address the need for synchronous spatiotemporal feature extraction in video-based dynamic gesture recognition, Molchanov et al. [**?**] first proposed extending traditional 2D CNN models, which can only extract image features, to three-dimensional models capable of synchronous spatial and temporal feature extraction, thereby effectively capturing motion information in videos. Wudi et al. [**?**] proposed a multimodal sign language recognition method using a two-column deep network: the first column employed 3D convolutional networks for motion feature extraction from video data, while the second column used deep belief networks (DBN) for recognition using skeleton data. Finally, they effectively fused the classification results from both sub-networks through voting to achieve a Jaccard Index score of 0.88 on the Montalbano gesture recognition competition dataset [**?**]. Pigou et al. [**?**] stacked 3D convolutional neural networks and recurrent neural networks (RNN), improving the Jaccard Index score on the Montalbano dataset to 0.916.

Although current deep network architectures have achieved good results in motion feature extraction and classification for gestures, video-based dynamic sign language recognition remains limited by the processing of long image sequences. Given the excellent performance of bidirectional long short-term memory (Bi-LSTM) networks in natural language processing tasks, this paper proposes a novel deep learning architecture called WRN-BCLSTM for long-sequence sign language recognition. The architecture first uses a 3D convolutional neural net-

work as a video feature extractor, feeding its fixed-length short spatiotemporal features into a multilayer bidirectional spatial LSTM for further encoding to form long-term correlation information. Then, a wide residual network (WRN) precisely represents the spatiotemporal information of long video sequences. Finally, through a fusion strategy, the model combines the classification errors of individual classifiers to achieve higher accuracy.

In summary, the innovations of this paper are: (a) designing 3D convolutional layers suitable for short-term feature extraction and bidirectional spatial LSTM layers suitable for long-term information encoding, enabling the model to maximally utilize video spatiotemporal features for classification; (b) designing residual modules for feature selection based on residual network principles, widening the convolutional kernel width of residual blocks while reducing corresponding layers, thereby expanding the spatial feature selection range and effectively solving gradient attenuation and uneven gradient problems in deep networks; and (c) proposing an effective fusion strategy for homogeneous data, achieving compensation for individual classifier errors when data is missing, resulting in higher classification accuracy.

---

1 Deep Network Architecture

To address dynamic gesture recognition, we propose a deep architecture that integrates 3D convolutional neural networks, bidirectional spatial LSTM networks, and wide residual modules, as illustrated in [Figure 1: see original paper].

First, videos containing gesture samples are processed into continuous image sequences of uniform length as model input. The 3D convolutional neural network then extracts features from the image sequences. After spatial and temporal feature data are synchronously parsed by the bidirectional spatial LSTM, long-term dynamic feature connection units are formed and input as two-dimensional tensors to the residual network. Following processing by residual layers, the features are finally fed into a softmax classifier, which outputs gesture sample predictions as a vector. Each dimension value in the vector represents the confidence that the current gesture belongs to a particular category.

1.1 Data Preprocessing

Due to the fully connected layers in deep network architectures, input data must have consistent dimensions. Therefore, data must first be unified in the temporal dimension. To accurately capture features representing gesture meanings, this paper employs the window sliding method [**?**], selecting 32 as the baseline frame number for each gesture video. For videos with more than 32 frames, irrelevant image sequences at both ends are removed, retaining only the middle key frames. For videos with fewer than 32 frames, interpolation is performed by selecting certain middle frames proportionally. Through this preprocessing, motion path

information in videos is preserved. The specific preprocessing procedure is as follows:

a) **Temporal segmentation of gestures**: As shown in [Figure 2: see original paper], the window sliding method standardizes video length to a fixed size (e.g., 32 frames). If the captured video length exceeds 32 frames, redundant frames at both ends are deleted; otherwise, certain frames are repeated.

b) **Spatial segmentation of gestures**: Each frame image is cropped to 112×112 pixels according to the human body region range, resulting in videos with uniform resolution.

1.2 3D Convolutional Neural Network Module

The 3D convolutional module structure used in this paper is shown in [Figure 3: see original paper]. The network input consists of 32 consecutive frames, with each frame having a spatial dimension of 112×112. The 3D convolution Conv3D-1 uses a kernel size of 3×3×3 with 64 features. Each 3D convolution kernel shares the same weight coefficients, and after convolution, the input data yields 64 feature maps of size 112×112×32. Similarly, Conv3D-2, Conv3D-3, and Conv3D-4 layers have 128, 256, and 256 convolution kernels respectively, all maintaining a size of 3×3×3. The pooling layer Pooling1 performs 2×2 downsampling only on spatial dimensions, while Pooling2 performs synchronous 2×2×2 sampling on both temporal and spatial dimensions. After three convolutional and two pooling operations, 256 feature maps of size 28×28×16 are obtained. Following each convolutional layer is a batch normalization (BN) layer, which normalizes the data distribution of each mini-batch to have zero mean and unit variance during gradient computation before feeding it to the next layer. Using batch normalization allows the initial learning rate to be set to a relatively large value, improving convergence speed.

Convolutional neural networks are a special type of feedforward neural network with three important characteristics—local connectivity, pooling, and weight sharing—that make them highly suitable for image data processing. Local connectivity ensures sparse connections between layers, greatly reducing model parameter scale. Pooling selects a specific value from a small region as output, thereby reducing feature dimensions. Weight sharing enables neurons within the same receptive field to have identical parameter values, further simplifying network structure and avoiding overfitting. Additionally, the stacking of convolution and pooling operations endows CNNs with a certain degree of translation, scaling, and distortion invariance [**?**].

Although traditional 2D CNNs possess strong feature extraction capabilities for image data, they tend to lose motion information between feature targets when handling video tasks due to the temporal dimension being converted into long frame sequences. To address this issue, this paper adopts a novel 3D CNN structure to improve upon traditional 2D CNN. The 3D convolution is defined

as follows:

$$v_{ij}^{xyz} = \sigma\left(b_{ij} + \sum_{m}\sum_{p=0}^{P_i-1}\sum_{q=0}^{Q_i-1}\sum_{r=0}^{R_i-1} w_{ijm}^{pqr} \cdot v_{(i-1)m}^{(x+p)(y+q)(z+r)}\right)$$

where v_{ij}^{xyz} represents the output of the 3D convolution operation, x, y, z denote the three dimensions of the input sample video data, with superscripts x and y representing spatial dimensions and z representing the temporal dimension. P_i, Q_i, R_i represent the convolution operation values in the three dimensions. Subscripts indicate the m-th feature map in the i-th layer, w_{ijm}^{pqr} is the weight value connecting the convolution kernel to the previous m-th feature map at coordinates (p, q, r), and b_{ij} is the bias parameter for the j-th feature map in the i-th layer. $\sigma(\cdot)$ is a continuous nonlinear activation function introduced to enhance the expressive capability of the structure.

Since the derivative value ranges of traditional sigmoid and hyperbolic tangent (tanh) activation functions are both less than 1, gradients continuously attenuate during backpropagation through each layer. Consequently, when network structure deepens, the vanishing gradient problem emerges. To align with biological neuron mechanisms, this paper uses the rectified linear unit (ReLU) as the activation function:

$$\mathrm{ReLU}(X) = \max(0, X)$$

When the input X value is less than or equal to 0, the output is forced to 0; when the input value exceeds 0, it remains unchanged. This introduces sparsity to the output, thereby accelerating network convergence.

1.3 Bidirectional Spatial LSTM Network Module

The goal of dynamic gesture recognition is to extract spatiotemporal visual information from video sequences. However, the temporal complexity of video events poses challenges for recognition tasks. Given the tremendous success of LSTM in handling complex temporal tasks in natural language processing, this paper explores using LSTM to recursively learn long-term dynamic features from input videos.

As a variant of recurrent neural networks, LSTM relies on memory cells to record all historical information up to the current moment in a sequence. It uses input gates i_t, forget gates f_t, and output gates o_t to control gradient propagation along the temporal dimension, enabling the mapping of input sequence x to hidden node sequence h and recursively learning complex temporal correlation information from dynamic features in input sequences. However, traditional LSTM in natural language processing treats one-dimensional vectors as processing objects, primarily learning temporal features after text vectorization.

Directly applying this structure to video classification tasks inevitably results in loss of image spatial location information.

At the 2015 NIPS conference, Shi et al. [**?**] proposed Convolutional LSTM (ConvLSTM), which can directly operate on two-dimensional tensors, effectively overcoming spatial information loss during temporal transmission and achieving success in video event analysis. Based on this structure, we designed a bidirectional spatial LSTM, as shown in [Figure 4: see original paper].

Two unidirectional ConvLSTM units connected together form a bidirectional (Bi-ConvLSTM) memory unit. Each Bi-ConvLSTM memory unit contains spatial and temporal inputs from the 3D convolution module. The computational structure of a Bi-ConvLSTM unit can be described as:

$$
\begin{aligned}
i_t &= \sigma(W_{xi} * x_t + W_{hi} * h_{t-1} + W_{ci} \circ c_{t-1} + b_i) \\
f_t &= \sigma(W_{xf} * x_t + W_{hf} * h_{t-1} + W_{cf} \circ c_{t-1} + b_f) \\
o_t &= \sigma(W_{xo} * x_t + W_{ho} * h_{t-1} + W_{co} \circ c_t + b_o) \\
c_t &= f_t \circ c_{t-1} + i_t \circ \tanh(W_{xc} * x_t + W_{hc} * h_{t-1} + b_c) \\
h_t &= o_t \circ \tanh(c_t)
\end{aligned}
$$

where x_t represents the current moment input, h_{t-1} represents the output at moment $t-1$, both stored as two-dimensional tensors. $*$ denotes convolution operation, $\circ$ denotes Hadamard product. i_t, f_t, and o_t represent input gate, forget gate, and output gate respectively. W and b terms represent input weights, recurrent weights, and bias terms for the three gate structures. i_t determines how much new information is added to a memory unit, f_t controls how much information each memory unit needs to forget, and o_t controls how much information each memory unit outputs. σ is the sigmoid nonlinear function, constraining gate element values between [0,1]. tanh is the hyperbolic tangent nonlinear function with range [-1,1]. c_t is the core memory unit controlling which information will be preserved, consisting of two parts: information retained from the previous memory unit c_{t-1} after the forget gate, and information retained from input data after modulation.

After spatiotemporal information from image sequences passes through bidirectional spatial LSTM transmission, effective global fusion is achieved. Compared to unidirectional LSTM, bidirectional LSTM networks can better capture global video information, thus yielding superior prediction results.

1.4 Wide Residual Network Structure Design

Following encoding by the stacked 3DCNN and Bi-ConvLSTM architecture, gesture videos are transformed into two-dimensional tensor features containing substantial spatiotemporal information. Deep convolutional neural networks have demonstrated excellent performance in image classification tasks in recent years. Particularly after 2015, advances in network models such as AlexNet,

GoogleNet, VGG, and ResNet enabled continuous breakthroughs in deep learning architectures for image classification. Based on this, this paper combines the outstanding deep residual architecture concept for innovative attempts at efficient and accurate automatic learning and classification of two-dimensional image features.

Theoretically, model capacity and feature discriminative capability can continuously improve as network depth increases. However, extensive experimental results indicate that simply increasing network depth introduces the gradient vanishing problem—excessively deep network structures easily lead to non-convergent training and consequently reduced recognition rates. To address this issue, He et al. [**?**] proposed building deep residual network structures using shortcut connections (SC).

Let $H(x)$ represent the optimal solution mapping of a deep neural network after inputting sample x. Traditional convolutional neural networks directly fit $H(x)$, whereas deep residual networks aim to fit the residual mapping $F(x) = H(x)-x$. Since x is the input source image, fitting $F(x)$ is equivalent to fitting $H(x)$. Under this condition, the original optimal solution mapping is expressed as $H(x) = F(x) + x$, as shown in Figure 5: see original paper.

Deep residual networks use shortcut connection structures to skip 2 or 3 convolutional layers, mapping themselves to the sum of stacked layer outputs and convolutional layer outputs. Clearly, optimizing a network to fit a determined function $F(x)+x$ is much easier than approximating an optimal function $H(x)$. The shortcut connection approach does not significantly increase model parameters or computational complexity. In other words, deep residual structures can be superimposed onto existing deep models without changing the original architecture, making it possible to train better-performing, deeper network models.

Nevertheless, even with identity mapping, residual networks suffer from the drawback that multiple residual modules share limited gradient information flow as depth increases—meaning only a small portion of residual module parameters get updated. To solve this problem, this paper incorporates the wide residual module concept proposed by Sergey et al. [**?**], using a shallow-and-wide structure instead of deep-and-narrow residual network modules. Subsequent experiments demonstrate that appropriately increasing residual module width improves residual network performance more than simply increasing network depth, as wider networks expand feature selection range and enhance feature coupling capability.

As shown in [Figure 5: see original paper], the wide residual network (WRN) module in this paper further extracts spatial features from the two-dimensional tensors output by Bi-ConvLSTM. The wide residual module has a total of 16 layers, consisting of 4 residual groups (Conv1, Conv2, Conv3, and Conv4). The width of residual groups is determined by widening coefficient k, with $k = 4$ in this paper. Each residual group contains $N = 4$ residual blocks. The first and second residual groups Conv1 and Conv2 correspond to Figure 5: see

original paper, while the third and fourth groups Conv3 and Conv4 correspond to Figure 5: see original paper, which performs spatial pooling in the first layer. Consequently, the number of feature maps in each layer is 8×4, 16×4, 32×4, and 64×4 respectively, effectively reducing residual module depth while widening convolutional kernel count without increasing model parameters.

---

2 Model Optimization

Model optimization can be understood as preliminary performance evaluation of the model using training and validation samples, followed by selecting appropriate hyperparameters to train an optimal decision model for final testing.

2.1 Loss Function and Regularization

Network architecture optimization is achieved through loss function calculation. This paper uses softmax in the output layer, with classification layer outputs calculated as:

$$P(y=i|x,\theta)=\frac{\exp(s_i)}{\sum_{k=1}^{K}\exp(s_k)}$$

where s_i represents the output of the i-th neuron, θ represents model parameters, and P represents the probability of a gesture corresponding to a certain category output by the network. For multi-classification computation, the loss function uses negative log-likelihood to reflect differences between network outputs and actual gesture labels:

$$L_D(\theta)=-\frac{1}{D}\sum_{i=1}^{D}\log P(y_i|x_i,\theta)$$

where x_i is the feature representation of input samples, y_i denotes target category labels, θ represents model parameters to be optimized, and D indicates the number of samples in a mini-batch. Network optimization is a process of continuously reducing error L_D by modifying parameters θ.

To address overfitting, this paper adds a regularization term to the softmax loss based on empirical risk minimization:

$$L=L_0+\lambda\|\theta\|_2^2$$

The first part L_0 corresponds to the original loss function, while the second part is an L2 norm regularization term that reduces parameter optimization space to avoid overfitting. λ is the regularization coefficient controlling the constraint strength on the loss function, with its value selectable through cross-validation.

2.2 Parameter Optimization

Parameter optimization involves backpropagating errors calculated by the loss function to compute gradients for each layer' s parameters. This paper uses an improved gradient descent optimization algorithm for neural network parameter updates:

$$\nabla_t = \langle \frac{\partial L_{\text{batch}}}{\partial \theta_t} \rangle$$
$$v_{t+1} = \mu v_t - \varepsilon \nabla_t$$
$$\theta_{t+1} = \theta_t + v_{t+1}$$

where ∇_t represents the gradient of loss function L_{batch} with respect to previous iteration parameters θ_t after training with one batch of data. The t-th iteration parameter update depends on the update occurring at the $(t-1)$-th iteration. ε is the learning rate; due to batch normalization in the network structure, the initial learning rate can be set to a relatively large value. To prevent overfitting, if the loss function reduction rate fails to meet expectations during iteration, corresponding weight decay is applied to ensure parameter update magnitude continuously decreases, biasing the learning process away from complex decision surfaces. μ is the momentum term representing accumulated parameter adjustment inertia from current iteration, set to 0.9. In early iterations, previous gradients accelerate training; in later iterations near convergence, when update directions are essentially opposite, gradients gradually diminish. This parameter update rule is similar to stochastic gradient descent (SGD), but differs in that it calculates momentum-based gradient v_t for weight updates, whereas SGD simply computes current weight gradients, resulting in significantly faster convergence.

2.3 Multi-Modal Fusion

With limited training samples, feature fusion has proven to be an effective means to further improve recognition performance [**?**]. As shown in the structural diagram [Figure 1: see original paper], this paper uses a two-column deep structure to extract different features from input videos. Each sub-network uses different data formats as input, yielding varying recognition effects. During testing, the fusion model combines estimated class probabilities from two sub-networks to calculate final gesture classification output:

$$P(C|x) \propto a \cdot P_I(C|x) + (1-a) \cdot P_{II}(C|x)$$

Here, class membership probabilities from different sub-networks are obtained via softmax function. a represents a weighting coefficient determined through cross-validation during training, controlling each sub-network' s contribution to final membership probability $P(C|x)$. Generally, a values are very close to 0.5.

Based on different data input formats, this paper trains two WRN-BCLSTM models separately and fuses their output probabilities, achieving robustness, high real-time performance, and high accuracy.

---

3 Experimental Results and Analysis

The experimental environment is as follows: Operating system: 64-bit Ubuntu 16.04 LTS; CPU: Intel Core i7-6700K octa-core; GPU: Nvidia GeForce GTX 1070 with 11264M memory; RAM: 32 GB DDR4; Framework: TensorFlow. Datasets used include the Sign Language Video in Museums (SLVM) dataset, ChaLearn Looking at People 2014 Gesture datasets (Montalbano), and Sheffield Kinect Gesture (SKIG) public datasets.

3.1 SLVM Dataset Results

Data is a crucial foundation and prerequisite for gesture recognition research. However, most current public datasets lack effective and accurate labeling or are stored in single data formats. To meet the demands of long-sequence dynamic sign language recognition research, this paper designed a multi-modal homogeneous signal data acquisition platform and established a high-frequency sign language vocabulary dataset used by deaf individuals during museum visits.

In the data acquisition module, to effectively suppress illumination and scene noise interference, this paper abandoned the conventional approach of using RGB images as training examples. Instead, we developed a multi-modal data acquisition system called Gestures Recorder based on Kinect V2 for Windows. As shown in [Figure 6: see original paper], the system synchronously saves features from infrared images, contour images, and skeleton data, collecting 20 categories of dynamic sign language vocabulary with a total of 6,800 samples: 5,100 training samples, 850 validation samples, and 850 test samples. Video resolution is 512×424, forming a complete sign language database.

This paper employs transfer learning to shorten model training time. Transfer learning shares parameters learned from the source task domain and generalizes them to similar learning tasks to improve model recognition rates. For dynamic gesture recognition, training deep network architectures from scratch may fail to achieve expected results due to insufficient data samples. To reduce training time, this paper adopts two transfer learning strategies: First, module transfer—migrating the first 9 layers of the 3DCNN module pretrained on the large-scale video dataset IsoGD (ChaLearn LAP 2017 RGB-D isolated gesture dataset) from the network in reference [**?**] to the current model for superposition with Bi-ConvLSTM and WRN modules. Since the 3DCNN layers extract edges, colors, and short spatiotemporal features that have certain commonalities in video classification, these first 9 layers' parameters can be fixed and fine-tuned during transfer. Second, data transfer—based on the determined 3DCNN+Bi-ConvLSTM+WRN architecture, classification learning is

performed on the large-scale IOSGD dataset to obtain a pretrained network model, which is then transferred to the SLVM dataset by replacing the classification output layer and fine-tuning parameters. The learning rate is set using a uniform distribution strategy, starting at 0.05 and multiplied by 0.1 after 1,920 iterations, with batch size set to 8. With GPU acceleration, one epoch completes in 2 hours, and the network converges well after 12 epochs.

[Figure 7: see original paper] shows the network's recognition accuracy curves on the SLVM dataset, with training iteration count on the horizontal axis and accuracy on the vertical axis. The best recognition rate reaches 98.3%. compares our method with the best published methods on the SLVM dataset. Results demonstrate that our approach offers significant advantages over the model in [**?**]. First, 3DCNN shares network layer parameters through convolution operations, effectively reducing network parameters while extracting local and even global spatial feature information from image sequences. Bi-ConvLSTM not only possesses the adaptive memory and anti-forgetting capabilities of traditional LSTM but also pays greater attention to spatial information in image sequences during temporal sequence learning. Consequently, our proposed model achieves a qualitative improvement over the previous best recognition accuracy on the SLVM dataset. Additionally, using ReLU activation functions provides stronger generalization capability, while L2 regularization helps avoid overfitting.

3.2 Montalbano Gesture Dataset Results

To verify algorithm effectiveness on large-scale datasets, this paper selected the Montalbano dataset for experimental comparison and analysis. This sign language dataset, recorded with depth cameras, aims to achieve user-independent dynamic gesture recognition based on multimodal data. The gesture recognition competition was held in 2014, containing 20 common Italian gesture expressions. The dataset comprises 13,858 gesture sequences performed by 27 individuals (9,494 training samples, 1,722 validation samples, and 2,642 test samples). Each multimodal sample includes traditional RGB images, depth images, skeleton data, and contour images. Detailed information about this public dataset can be found in reference [**?**].

This paper follows Montalbano competition rules, comprehensively evaluating algorithm performance using Jaccard Index scores. The official competition evaluation method calculates:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

where A and B represent the ground truth boundary and algorithm-predicted boundary for a dynamic gesture, respectively. The final algorithm score on the dataset is the mean Jaccard Index across all test samples:

$$\text{mean}J = \frac{1}{N \cdot S} \sum_{s=1}^{S} \sum_{n=1}^{N} J_{s,n}$$

Here, $N = 20$ represents the number of gesture categories, and S denotes the frame sequence length of test samples.

Considering the time-consuming nature of training complex deep networks, especially on large datasets like Montalbano, this paper uses the network architecture trained on the IsoGD dataset as an initialization model for parameter tuning on Montalbano. The initial learning rate is set to 0.01, decaying to 1/10 every 2,500 iterations, with batch size set to 8. Experiments show that with limited data samples, transferring pretrained models not only saves training time but also significantly improves recognition accuracy.

As shown in [Figure 8: see original paper], the printing frequency on the Montalbano dataset displays one stage result every 95 iterations. Due to the pretrained model transfer, the network loss value decreases rapidly, saving time for further optimization. compares various algorithms on the Montalbano dataset. The first three works [**?**] use manual feature extraction methods, demonstrating that deep learning offers enormous advantages over manual feature extraction. Method [**?**] is our previous approach using 3DCNN with data fusion technology, proving that effective data fusion can improve overall recognition rate by nearly 5% compared to single network models. Method [**?**] uses a probability-based autoencoder with restricted Boltzmann machines (RBM) for skeleton data and CNN for RGB-D images, effectively combining static classifier advantages with dynamic sequence processing via hidden Markov models. Method [**?**] proposes learning audio+video modal features with neural networks, proving multimodal fusion learns better features than single modalities, achieving first place in that competition. Method [**?**] borrows RNN' s success in natural language processing, first applying CNN+RNN models to gesture recognition, achieving qualitative improvement in accuracy and proving RNN' s excellent performance on sequential samples.

Similar to method [**?**], our approach uses 3D convolutional neural networks to automatically extract short-term features from videos. The difference lies in combining spatial convolutional LSTM recurrent networks to preserve spatial correlation information between video sample sequences while learning long-sequence features, yielding more precise classification results. Recognition results using depth video are slightly higher than RGB video because RGB images are more susceptible to environmental illumination and complex backgrounds. Fusion strategy results outperform any single feature, proving the effectiveness of our proposed feature fusion method.

3.3 SKIG Gesture Dataset Results

To further verify algorithm effectiveness for dynamic gesture classification, this paper selected the Sheffield Kinect Gesture Dataset (SKIG) for experimental comparison and analysis. Established by Ling Shao et al. [**?**], the dataset contains 2,160 independent gesture samples (1,080 RGB and 1,080 depth videos). Each sample contains one independent dynamic gesture across 10 categories: circle, triangle, up-down, right-left, wave, Z, cross, come here, turn around, and pat. To reflect data diversity in real-world complex backgrounds and illumination conditions, the dataset was recorded by 6 performers under three backgrounds (white, wood grain, and newspaper with text) and two lighting conditions (strong and weak light) using three different postures (fist, single finger, and palm). Although these samples appear as the same category to human eyes, different positions, backgrounds, and lighting conditions represent different neuron responses for network models, making this dataset valuable for verifying algorithm robustness across modes and execution environments.

The experimental design on this public dataset follows the same principle as reference [**?**]: 3-fold cross-validation, where each time samples from 4 performers are used for training and validation sets, and samples from the remaining 2 performers serve as the final test set. To leverage commonalities between different learning tasks and shorten training time, experiments use the network trained on the Montalbano dataset as an initialization model, changing output layer nodes to match SKIG dataset parameters, and fine-tuning two sub-networks separately on RGB and depth videos. The initial learning rate is 0.01, decaying to 1/10 every 150 iterations.

As shown in [Figure 9: see original paper], with batch size set to 8, the network trains for 10 epochs, printing results every 9 iterations. Due to the transfer learning strategy, the model demonstrates strong learning capability early in training, with loss values decreasing rapidly. After 4 epochs, loss values stabilize, and after 8 epochs, they approach 0, indicating near convergence. Recognition accuracy shows a similar trend: test accuracy rises quickly early in training and stabilizes later, reaching 99.3%.

lists accuracies achieved by various published methods on the SKIG test set. As a baseline method for this public dataset, reference [**?**] proposed an adaptive method fusing RGB and depth data, using graph-constrained genetic algorithms for optimization to effectively classify dynamic gestures. Reference [**?**] used a dynamic gesture recognition method based on deep fusion of shape and spatiotemporal features, combining two data feature descriptors with linear SVM classifiers to achieve 98.4% accuracy on SKIG. Reference [**?**] first applied the combination of convolutional and recurrent neural networks to this dataset: 3D CNN extracted 3D spatiotemporal motion information, which was then used by RNN to capture long-sequence spatiotemporal features, achieving high-precision dynamic gesture classification. Reference [**?**] first applied spatial LSTM models to gesture classification, using spatial pyramid pooling (SPP) before the classi-

fier to extract and aggregate spatial information from feature maps of different sizes, effectively saving training time. Our method uses bidirectional spatial LSTM to maximally utilize video spatial information for fusing long-sequence information and employs wide residual networks for accurate classification of video-converted two-dimensional tensors, achieving further accuracy improvement.

4 Conclusion

This paper designs a deep model called WRN-BCLSTM that integrates wide residual modules and bidirectional spatial LSTM networks for dynamic gesture recognition tasks. The model uses 3D convolutional neural networks to extract spatial and short-term dynamic features from video streams, employs bidirectional spatial LSTM networks to fully capture contextual long-term information from 3DCNN outputs, and introduces wide residual modules in the latter part of the model. This effectively solves feature attenuation and reuse problems in traditional deep stacked residual modules, improving model discriminative capability.

Experiments on three public dynamic gesture datasets validate the model' s effectiveness. Results reveal that: (1) Inputting two data streams separately into the network and fusing their outputs is effective, with fusion strategies significantly improving recognition performance over any single data stream feature. (2) ConvLSTM not only possesses temporal modeling capabilities of traditional LSTM but also characterizes spatial local features like CNN, making it more suitable for spatiotemporal information processing. (3) Appropriately increasing wide residual module width improves residual network performance more than simply increasing network depth, with wide residual networks also training faster. (4) With limited data samples, transfer learning strategies not only shorten training time but also yield better generalization.

Deep learning technology progressively abstracts and transforms features from raw data space, ultimately classifying using high-level features, greatly improving dynamic gesture recognition accuracy. However, many unknown factors in deep models remain to be explored, such as designing residual shortcut connections for 3D convolutional modules. Additionally, other deep learning network structures like attention model-based deep networks have shown great advantages in human action understanding, representing our future research directions.

References

[1] Sharma R, Pavlovic V I, Huang T S. Toward multimodal human-computer interface [J]. Proceedings of the IEEE, 1998, 86(5): 853-869.

[2] Dardas N H, Georganas N D. Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques [J]. IEEE Trans on Instrumentation & Measurement, 2011, 60(11): 2532-2541.

[3] Parcheta Z, Martínez-Hinarejos C D. Sign language gesture recognition using hmm [C]// Proc of Iberian Conference on Pattern Recognition and Image Analysis. Berlin: Springer, 2017: 419-426.

[4] Radu-Daniel V. Beyond features for recognition: human-readable measures to understand users' whole-body gesture performance [J]. International Journal of Human-Computer Interaction, 2017, 12(23): 1-16.

[5] Simon F, Helena M, Pushmeet K, et al. Instructing people for training gestural interactive systems [C]// Proc of SIGCHI Conference on Human Factors in Computing Systems. New York: ACM Press, 2012: 1737-1746.

[6] Cao Jie, Zhao Xiulong, Wang Jinhua. Dynamic gesture recognition approach based on RGB-D information [J]. Application Research of Computers, 2018, 35(7): 2228-2232.

[7] Zhang Beiwei, Wu Qi, Liu Guanghui. Method for recognizing gesture of traffic police based on DTW algorithm [J]. Application Research of Computers, 2017, 34(11): 3494-3499.

[8] Mohanty Aparna, Sahay Rajiv R. Understanding Indian classical dance by recognizing emotions using deep learning [J]. Pattern Recognition, 2018, 7(79): 97-113.

[9] Hyeon Chul Moon, Anna Yang, Jae-Gon Kim. CNN-Based Hand Gesture Recognition for Wearable Applications [J]. Journal of Broadcast Engineering, 2018, 3(23): 246-252.

[10] Molchanov Pavlo, Gupta Shalini, Kim Kihwan, et al. Hand gesture recognition with 3D convolutional neural networks [C]// Computer Vision and Pattern Recognition Workshops. New York: IEEE, 2015: 1-7.

[11] Wu Di, Lionel Pigou, Pieter-jan Kindermans, et al. Deep Dynamic Neural Networks for Multimodal Gesture Segmentation and Recognition [J]. IEEE Trans on Pattern Analysis & Machine Intelligence, 2016, 38(8): 1693-1706.

[12] Wan Jun, Sergio Escalera, Gholamreza Anbarjafari, et al. Results and Analysis of ChaLearn LAP Multi-modal Isolated and Continuous Gesture Recognition, and Real Versus Fake Expressed Emotions Challenges [C]// IEEE International Conference on Computer Vision Workshops. New York: IEEE, 2018: 3189-3197.

[13] Zhu Guangming, Zhang Liang, Shen Peiyi, et al. Multimodal Gesture Recognition Using 3D Convolution and Convolutional LSTM [J]. IEEE Access, 2017, 5(99): 4517-4524.

[14] Tu Zhigang, Xie Wei, Qin Qianqing, et al. Multi-stream CNN: Learning representations based on human-related regions for action recognition [J]. Pattern

Recognition, 2018, 2(79): 32-43.

[15] Shi Xinjian, Chen Zhourong, Wang Hao, et al. Convolutional LSTM Network: a machine learning approach for precipitation nowcasting [C]// Proc of International Conference on Neural Information Processing Systems, Massachusetts: MIT Press, 2015: 802-810.

[16] He Kaiming, Zhang Xiangyu, Ren Shaoqing, et al. Deep Residual Learning for Image Recognition [C]// Proc of IEEE Conference on Computer Vision and Pattern Recognition. New York: IEEE Computer Society, 2016: 770-778.

[17] Guo Dan, Zhou Wengang, Li Houqiang, et al. Online Early-Late Fusion Based on Adaptive HMM for Sign Language Recognition [J]. ACM Trans on Multimedia Computing Communications & Applications, 2018, 14(1): 1-19.

[18] Necati Cihan, Ahmet Alp kindiroglu, Lale Akarun. Gesture Recognition Using Template Based Random Forest Classifiers [C]// Proc of Computer Vision. Zurich: Springer International Publishing, 2014: 579-594.

[19] Chang, Juyong. Nonparametric Gesture Labeling from Multi-modal Data [C]// Proc of Computer Vision Workshops. Zurich: Springer International Publishing, 2014: 503-517.

[20] Camille Monnier, Stan German, Andrey Ost. A Multi-scale Boosted Detector for Efficient and Robust Gesture Recognition [C]// Proc of Computer Vision Workshops. Zurich: Springer International Publishing, 2014: 491-502.

[21] Liang Zhijie, Liao Shengbin, Hu Bingzhang. 3D convolutional neural networks for dynamic sign language recognition [J]. Computer Journal, 2018, 5(4): 1-13.

[22] Neverova Natalia, Wolf Christian, Taylor Graham, et al. ModDrop: Adaptive Multi-Modal Gesture Recognition [J]. IEEE Trans on Pattern Analysis & Machine Intelligence, 2016, 38(8): 1692-1706.

[23] Pigou Lionel, Van Den Oord Aaron, Dieleman Sander, et al. Beyond Temporal Pooling: Recurrence and Temporal Convolutions for Gesture Recognition in Video [J]. International Journal of Computer Vision, 2015, 5(3): 1-10.

[24] Li Liu, Ling Shao. Learning discriminative representations from RGB-D video data [C]// Proc of International Joint Conference on Artificial Intelligence. Palo Alto: AAAI Press, 2013: 1493-1500.

[25] Pavlo M, Yang Xiaodong, Gupta Shalini, et al. Online detection and classification of dynamic hand gestures with recurrent 3D convolutional neural networks [C]// Computer Vision and Pattern Recognition. New York: IEEE, 2016: 4207-4215.

[26] Zheng Jinqing, Feng Zhiyong, Xu Chao, et al. Fusing shape and spatio-temporal features for depth-based dynamic hand gesture recognition [J]. Multimedia Tools & Applications, 2017, 5(76): 1-20.

[27] Zagoruyko S, Komodakis N. Deep compare: a study on using convolutional neural networks to compare image patches [J]. Computer Vision & Image Understanding, 2017, 16(64): 1-9.

Note: Figure translations are in progress. See original paper for figures.