International Journal on Advanced Science Engineering Information Technology

Vol.14 (2024) No. 5
ISSN: 2088-5334

Integration of CNN and LSTM Networks for Behavior Feature Recognition: An Analysis

Teh Noranis Mohd Aris [a,*], Chen Ningning [a], Norwati Mustapha [a], Maslina Zolkepli [a]

[a] *Department of Computer Science, Universiti Putra Malaysia, Serdang, Selangor, Malaysia*
Corresponding author: [*]nuranis@upm.edu.my

***Abstract*— This study explores an integration model combining convolutional neural networks (CNNs) and long short-term memory networks (LSTMs) for behavior feature recognition. Initially, a straightforward three-dimensional deep CNN structure was introduced for behavior recognition, capturing static and dynamic characteristics, and analyzing the network's convergence speed. Subsequent experiments utilize the VGG16 CNN model, substituting the fully connected layer with global average pooling. Then, a comparative experiment was conducted on the MSRC-12 behavior dataset between the models. Due to the complexity of LSTM, a simpler GRU model with similar effectiveness was used for comparison. The experimental results showed that the GRU-CNN model performed best, outperforming other algorithms in the literature on the same dataset. Under the same experimental parameters, the GRU-CNN model converges significantly faster than the LSTM-CNN model, with speedier training speed. In addition, the best accuracy is achieved by adjusting the dropout and epoch. Due to cross-validation in this study, the GRU-CNN models achieved good experimental results when the hidden node dropout rate was 0.5. The epoch size had negligible impact on the GRU-CNN model. Still, the accuracy of the CNN and CNN-GRU models increased significantly with more epochs, further validating the effectiveness of the GRU-CNN model. These experiments also indicate that convolutional neural networks based on deep learning are superior to traditional machine learning methods for human behavior recognition. Using depth images instead of conventional images allows for better extraction of spatial features, and the integration with long short-term memory networks enhances the extraction of temporal features from sequences.**

***Keywords*—CNN; LSTM; behavior feature recognition; GRU.**

Manuscript received 15 Feb. 2024; revised 24 Jul. 2024; accepted 8 Aug. 2024. Date of publication 31 Oct. 2024.

I. INTRODUCTION

Behavior recognition is one of the hot topics in the field of computer vision [1], is widely applied in various areas such as intelligent security, autonomous driving, smart home, and patient monitoring. The main goal of behavior recognition is to identify the category of human actions in images, which involves preprocessing the data, detecting the moving human body, extracting features, performing recognition, and finally completing the classification [2]. Due to various challenges such as complex backgrounds, occlusions, lighting, intra-class and inter-class variations, and temporal changes, behavior recognition is currently still in the laboratory testing phase. Addressing these issues, this paper studies human behavior recognition based on deep learning's convolutional neural networks (CNN) and long short-term memory networks (LSTM) for solving time series problems [3], [4].

Manually crafted features often heavily rely on training data. In contrast, deep learning can autonomously learn features with high discriminative power from training samples, making classification more accurate [3], [5], [6]. Convolutional architectures can achieve stable latent representations at each layer, enable local interactions in space and time, and associate feature mappings with multiple consecutive frames of the previous layer, obtaining spatial structural information of internal frames and inter-frame-related information [6].

A. Deep 3D CNN Feature Extraction Model

Based on prior research, we developed a new deep 3D CNN model using an appropriate network size and topology. The designed 3D CNN model is built on the Keras framework as a general model, consisting of 4 convolutional layers, 2 pooling layers, 2 fully connected layers, and a SoftMax classification layer. The first two convolutional layers have sixty-four kernels each, padded using the "VALID". The latter two convolutional layers have 128 kernels each, padded using the "SAME", with a kernel size of 3×3 for convolutional layers, and the pooling layers use max pooling with a kernel size of 2×2×2 and a stride of 1. We use a Flatten layer to

unfold multi-dimensional input into one-dimensional output, employ Bayesian methods for parameter optimization, use a dropout layer [7], [8] to prevent overfitting, use the 'ReLU' function as the activation function, and use cross-entropy as the loss function, which has the advantage of using the Sigmoid function to avoid the learning rate reduction problem of mean squared error loss functions during gradient descent. The formula for the loss function 1 is as follows:

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{j=1}^{k} 1\{y^{(i)} = j\} log \frac{e^{\theta_j^T x^{(i)}}}{\sum_{i=1}^{k} e^{\theta_j^T x^{(i)}}}\right] \quad (1)$$

where $1\{\cdot\}$ is an indicative function with $1\{value\ true\} = 1$ and $1\{value\ false\} = 0$.

The structure of the deep 3D convolutional network designed in this paper is shown in Figure 1.

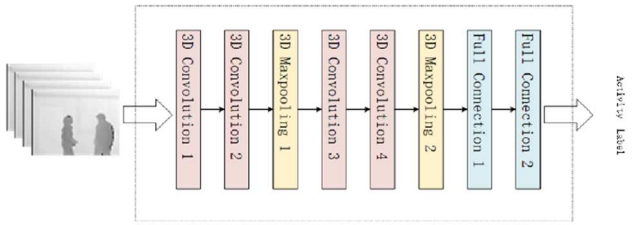

Fig. 1 Deep 3D Convolutional Network Structure

A 3D convolutional neural network can compute any value V on the jth feature map for layer i using the following equation 2:

$$V_{ij}^{xyz} = tanh\ tanh\left(b_{ij} + \sum_{m}\sum_{p=0}^{P_{i-1}}\sum_{q=0}^{Q_{i-1}}\sum_{r=0}^{R_{i-1}} w_{ijm}^{pqr} v_{(i-1)m}^{(x+p)(y+q)(z+r)}\right) \quad (2)$$

where p_i is the height and q_i is the width of the 3D convolution kernel, r is the size of the 3D convolution kernel in the time dimension, w_{ijm}^{pqr} is the channel connecting the values of the m th feature map point $(p,\ q,\ r)$ of the previous layer, and b_{ij} is the bias of the feature map. The method is generic and can be applied to different behavioral classes.

The "ReLU" function can speed up the convergence of network learning while making the output somewhat sparse and enhancing the classification ability of the network. The formula is as follows:

$$relu(x,y,t) = Conv(x,y,t) \cdots if \cdots Conv(x,y,t) > 0 \quad (3)$$

The first pooling layer is defined as shown in Equation 4.

$$Pool(x,y,t) = \big(ReLU(x,y,t)\big) \quad (4)$$

As the complexity of the network prediction model increases, the amount of training data required also increases sharply. We uniformly set the size of the convolution kernels to 3×3×3 across all layers. Choosing smaller filter sizes aims to reduce the parameter number for our model, making the training easier. Overfitting is a common issue, usually occurring when deep neural networks have too many parameters relative to the number of outcomes. If overfitting occurs, the predictive performance of the deep architecture will deteriorate. dropout is used to overcome overfitting, by generating numbers from 0 to 1 through a uniform function. From the experiment, the random dropout layers have significant achievement in many tasks related to recognizing human behavior. Bayesian optimization methods are used to select the hyperparameters of the proposed deep CNN architecture. During training the 3D CNNs model, we set the epoch as 1000, the weight decay rate as 0.0005, the momentum is 0.9, the batch size is 10, and the learning rate is 10^{-4}.

B. Performance Analysis based on the MSR-Action3D Dataset

An established 3D convolutional neural network processes original image sequences as input and makes predictions regarding class labels. The size of all image sequences has been adjusted to 64×48. To reduce the training time, we extract non-overlapping 16-frame segments from the depth image sequences at regular time intervals, with an input size of 3×16×64×48.

In the experiment, image sequences are divided into two groups of 8:2 ratio randomly, with 80% to be training and 20% to be testing. The experimental results are shown in Table I, the network using only depth information outperforms the previous methods with the highest accuracy of 95.63%. The experimental results indicate that the proposed deep 3D CNN structure is effective for human behavior recognition and has good classification performance. This fully demonstrates the feasibility of the proposed 3D convolutional neural network model for human behavior recognition. Furthermore, to demonstrate the effectiveness of the deep 3D CNN model proposed, we utilize a human interaction dataset for human behavior recognition.

TABLE I
CLASSIFICATION ACCURACY FOR DIFFERENT FEATURES

Method	Feature	Accuracy (%)
Wang [9]	Actionlet	88.20
Oreifej [10]	HON4D+Ddisc	88.90
Hossein [11]	LCSS+MIJA	91.20
Yang [12]	SNV	93.09
Lu [13]	Range Sample	95.63
Yong [14]	HBRNN	94.49
Shi [15]	PRNN	94.90
Our method	3D-CNN	96.88

II. MATERIALS AND METHOD

A. CNN Network Structure

Comparison experiments were first conducted under the VGG16 model, as shown in Figure 2. VGG16 consists of 13 convolutional layers and three fully connected layers, with the last layer being a SoftMax classification layer. All the activation functions of this network are using the ReLU

function [16]. The skeletal data is first fed into the conv2D convolutional layer and padding operation is performed to complement the zero operation on the image. VGG16 network is first convolved by two layers of 64 3×3×3 convolutional kernels, and after using the maximum pooling layer the length and width of the matrix are changed to half of its original width, which is then fed into the two convolutional layers. After convolution with 128 convolutional kernels, it is maximally pooled, with the 13 layers of convolutional layer and pooling layer after using the fully connected layer to unfold the data, and finally fed into the SoftMax layer to classify the features.

The fully connected layer is converted to global average pooling (GAP) based on the VGG16 network [17]. The global average pooling layer averages all the values of the feature maps by summing them up, i.e., one feature map corresponds to one output, which avoids the black box operation of the fully connected layer and prevents the network from overfitting. Figure 2 shows the CNN network structure. Figure 3 shows the global average pooling schematic. Figure 4 shows the convolutional network structure.

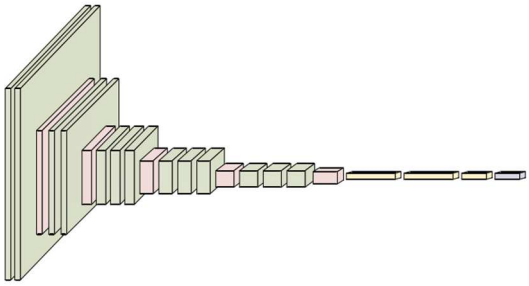

Fig. 2 VGG16 model diagram

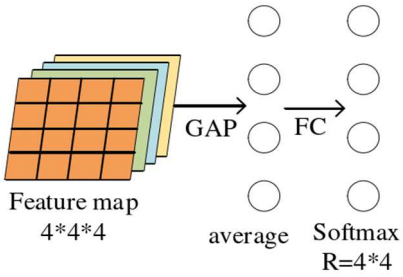

Fig. 3 Schematic of global average pooling

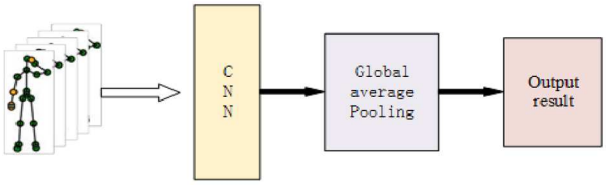

Fig. 4 CNN network structure

B. CNN-LSTM and LSTM-CNN Network Structures

In 2015, Donahue et al **Error! Reference source not found.** proposed adding an LSTM network model on top of the CNN network structure, referred to as the Long-term Recurrent Convolutional Network (LRCN). For comparative experiments, this paper adds a Long Short-Term Memory network to the original CNN network architecture, here using a Bidirectional Long Short-Term Memory network (BiLSTM) [19], [20] instead of LSTM [21], [22]. The bidirectional long and short-term memory network is shown in Fig. 5. The part in the box is the same as the unidirectional LSTM, the difference is that BiLSTM consists of forward LSTM and backward LSTM together, and the bidirectional LSTM can better capture the bidirectional motion feature relationship thus making full use of the information in the video sequences and understanding the human body behavior recognition from different perspectives.

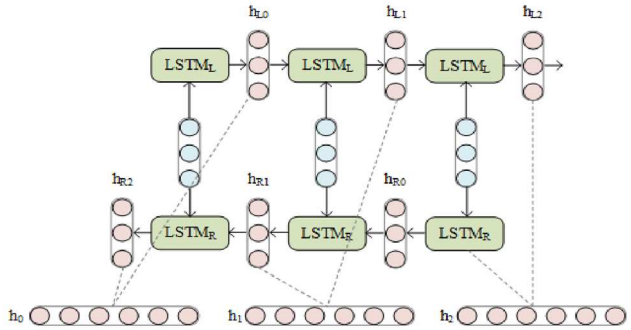

Fig. 5 Bidirectional long- and short-term memory network

The CNN-LSTM network model with the addition of a bidirectional LSTM is shown in Figure 6. The model consists of a convolutional neural network model as the initial layer, which is used to receive the sequence of skeletal frames to extract the local features of the image, and then after the convolutional network is input to the LSTM layer to extract the temporal information in the sequence.

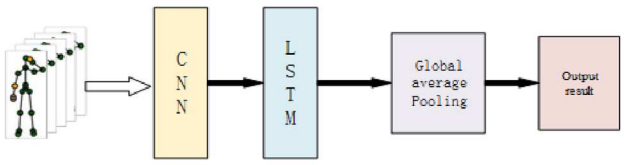

Fig. 6 CNN-LSTM network model

The model combines CNN and LSTM [23], [24]. CNN-LSTM model is compared with an LSTM-CNN model. The LSTM-CNN model is shown in Fig. 7. In a model for processing text first input the data to the long short-term memory network to extract the sequence information of the sentence, and then input to the convolutional neural network to capture some key information in the sentence, the model obtained good results in text processing. Therefore, this paper draws an idea of text classification to take the LSTM layer as the initial layer, and the skeletal sequence is input to the LSTM layer. Then the input from the LSTM layer is immediately followed by the input to the CNN model to extract local features. In the CNN-LSTM model, the LSTM layer only acts as a fully connected layer, which does not show the advantages of LSTM for this model. Whereas for the LSTM-CNN model, the initial layer LSTM acts like an editor to label each sequence of the input, which contains not only the information of the original labeling but also the output labeling of all other previous information. Subsequently the convolutional neural network will come in further to find the local features for better accuracy.

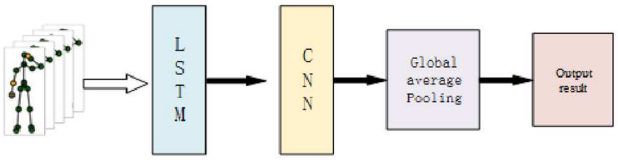

Fig. 7 LSTM-CNN network model

C. GRU-CNN Network Structure

Although LSTM networks have strong processing ability for data with long-time correlation changes, compared with GRU networks, GRU networks [25], [26] have fewer network parameters and shorter convergence time, which can better meet the real-time demand of human-computer interaction. Therefore, this paper proposes a GRU-CNN fusion network model, which first uses a GRU network to label the input time series, which contains not only the information of the original labeling, but also the historical information of the output labeling, and then carries out the feature extraction through a CNN network. The structure is richer than the features extracted by using CNN directly, which is helpful for us to get better recognition rate and robustness, and finally output to SoftMax by global pooling before classification.

The GRU-CNN model is shown in Figure 8. Distance features are suitable for input to GRU due to the abundance of spatial information, but the distance features lose a large amount of directional information. CNN is used to capture directional information. Due to the huge parameters of the vgg16 model, which consumes lots of time, Experiments have shown that the vgg16 models is easily overfitted due to the small size of the dataset. After multiple experiments, it was found that a three-layer convolution can achieve the optimal state. Therefore, this paper uses a three-layer convolutional layer to extract local information from the data. The model uses a framework of 3 bidirectional GRU layers. The first GRU layer takes the data of 20 skeletal key point frame sequences for gesture behavior recognition as input, which is the position vector of each skeletal key point. For the GRU, the output of the first layer is the input to the second layer, and the last layer's output is fed into the convolutional layer. Since the convolution kernel size affects the final feature extraction and ultimately the accuracy of behavior recognition, we adopt the idea of text classification here and set $p \times q$ is the convolution kernel, the number of skeletal key points in the convolution window is p.

By extracting the connections between adjacent skeletal key points, p captures the key information of human behavior and determines the category of the entire behavior, while q is the vector dimension of the image. Finally, the output of the convolutional layer is globally average pooled and then input into the fully connected layer. The dropout layer prevents overfitting and improves the generalization ability of the model. The results are then input into the SoftMax layer for classification. In this model, the GRU framework extracts the temporal sequence features of skeleton sequence, while the output of the GRU is the input for the CNN to obtain the spatial features of the skeleton.

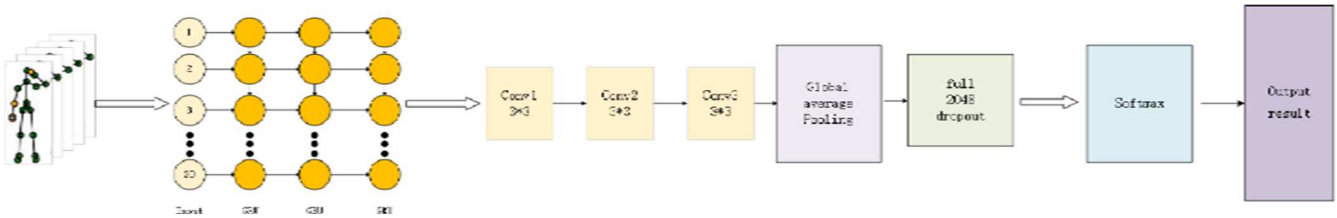

Fig. 8 GRU-CNN network model

III. Results and Discussion

A. MSRC-12 Dataset

The MSRC-12 [27] action dataset consists of various static images, retrieved from the Microsoft Research Institute's Cambridge University Computer Laboratory through Kinect motion-sensing devices.

B. Experimental Results

The GRU layer is the initial layer, with the sequence input. Then, the GRU output is input into the CNN model to extract local features. This paper applies 80% for the training and 20% for the testing. To confirm the feasibility of the model, we conducted several sets of comparative experiments. To illustrate the significant role of the GRU model, this paper also designs a CNN-GRU model and conducts experimental comparisons.

1) Conducting experiment on the TensorFlow and Keras deep learning platforms with an NVIDIA v100 graphics card on an Ubuntu system:

The Dropout rate was 0.5, the learning rate 0.001, and the epochs set to 10. The results are shown as follows:

TABLE II
COMPARISON EXPERIMENTS ON THE DATASET MSRC-12

Method	Year	Accuracy (%)
Cov3DJ [27]	2013	91.76
ConvNets[28]	2015	84.46
JTM [29]	2016	93.12
ASM-3[30]	2017	97.60
RF(N=23) +SW [31]	2017	98.37
hd-CNN [32]	2018	94.59
TPSMMs [33]	2019	96.53
CNN	Ours	98.5
CNN-LSTM	Ours	98.32
LSTM-CNN	Ours	99.6
CNN-GRU	Ours	98.3
GRU-CNN	Ours	99.8

The method we proposed is compared with previous methods in recent years, the results show that the model has achieved the highest recognition rate, which validates the feasibility and effectiveness of the model. In the CNN-GRU

model, the GRU only acts as a fully connected layer, which for this model does not bring out the advantages of GRU for processing time series. For the GRU-CNN model, the initial layer of GRU acts as an editor to label each input sequence. It captures not only the current labeling information but all previous output labeling information. After that, the CNN layer will use a richer representation of the original input to find local features, resulting in better accuracy.

2) Verifying the advantages of the GRU model over the LSTM model:

We conducted a comparison experiment between the GRU-CNN and the LSTM-CNN. In Figure 9, under the same experimental parameters, the convergence speed of the GRU-CNN model is significantly better than that of the LSTM-CNN model. In terms of the final training accuracy, there is not much difference. However, since the structure of GRU is simpler than LSTM, the parameters are less and easier to converge, and the training speed is faster.

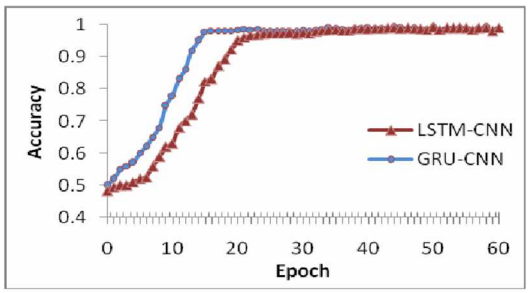

Fig. 9 Training accuracy of LSTM-CNN and GRU-CNN models

This experiment analyzes the impact of dropout and epoch settings on the model. The effects of different dropout parameter settings on the model experiment are shown in Figure 10. The role of the dropout is to keep the weights of the certain implicit layer nodes of the network temporarily inactive during model training to prevent model overfitting. Due to the cross-validation used in this paper, when the implied node discard rate is 0.5, all comparative models achieved better experimental results since the most randomly generated network structures are available when the discard rate is 0.5. When the dropout is 0.3, it will have the greatest impact on the CNN-GRU, reducing it by almost 50%. However, the accuracy of the CNN and GRU models improves. When the dropout is 0.9, all three models are reduced by 5-21%.

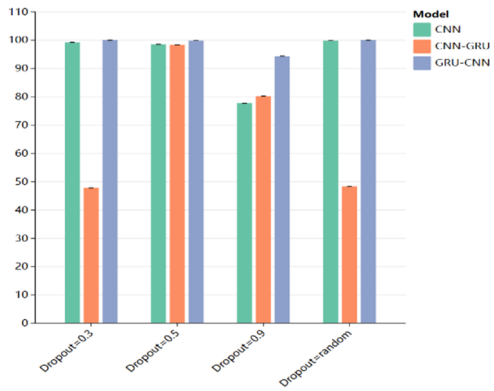

Fig. 10 Comparative experiments on the impact of dropout parameters on the model

From the experimental results, it has been proven that the GRU-CNN model proposed in this paper achieves satisfactory results with strong robustness under different dropout parameters. The effect of epoch on the model's recognition accuracy is shown in Figure 11, from the results we can see the epoch size has little effect on our model, but the accuracy of CNN and CNN-GRU models will greatly increase with the increase of epoch, verifying the effectiveness of the model.

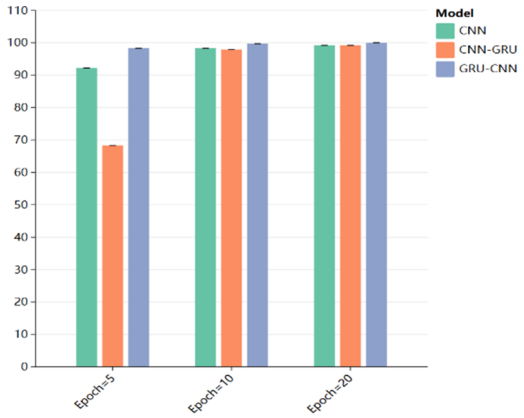

Fig. 11 Comparative tests of epoch on model accuracy

IV. Conclusion

First, a 3D convolutional neural network (CNN) architecture is designed, which can autonomously learn behavior recognition features from deep image sequences. The initial stage of the process involves the extraction of image features through the application of two convolutional layers. A maximum pooling layer then downscales these features. Subsequent inputs to another set of convolutional layers serve to refine the feature extraction. To prevent overfitting, these features are then processed by a pooling layer. The architecture includes two fully connected layers, which increase the model's complexity and improve its learning capability by enhancing non-linear representation. These layers are of significant importance in capturing static features and dynamic features. The objective is to extend the deep image sequence model to encompass structural interdependencies, utilizing deep hierarchical feature learning. The experimental results demonstrate the efficacy of the proposed 3D CNN structure in accurately identifying behavioral features. The next model to be explored is the combined model of CNNs and LSTMs. Initially, the established VGG16 CNN model is enhanced by implementing global average pooling to replace fully connected layers, which reduces the parameter count and mitigates overfitting. Subsequently, the LSTM network is integrated with CNN to facilitate spatiotemporal feature extraction. Comparative analysis of the CNN, CNN-LSTM, and LSTM-CNN models indicates that the LSTM-CNN model exhibits superior recognition accuracy. However, due to the extensive parameters and complexity of LSTM, we refine the LSTM-CNN model by substituting LSTM with its simplified variant, the Gated Recurrent Unit (GRU), which significantly reduces computational demands. Experimental results confirm that GRU achieves performance comparable to LSTM while exhibiting reduced complexity. By fine-

tuning the parameters, particularly through dropout and epoch adjustments, we determine the optimal settings, establishing the robustness of the GRU-CNN model. Our findings demonstrate that deep learning based on CNNs outperforms traditional machine learning in human behavior recognition. This is evidenced by the superior performance of deep images in spatial feature extraction, and the integration of LSTM networks enhancing temporal feature extraction.

Regarding the research in this paper, the next work in the research plan and the issues that need to be addressed can be divided into two parts. First, in the deep 3D convolutional neural network model mentioned in this paper, depth images are affected by dark objects, (semi-)transparent objects, specular reflection objects, and disparity. The quality of depth maps is closely related to hardware, which poses challenges in high power consumption and cost for processing depth images. Secondly, the GRU-CNN model designed in this paper has only been validated on one dataset, lacking suitable datasets for further processing. Moreover, using global average pooling instead of fully connected layers in the final model reduces the number of parameters. This poses challenges for using the model for other feature extraction tasks on different datasets or when the pre-tuned parameters are no longer suitable, making transfer learning difficult. Furthermore, the model designed in this paper is more practical for data with sequential features, and improvements are needed for behavior recognition datasets without such sequential characteristics.

ACKNOWLEDGMENT

We thank the Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, for providing financial support under project code 6236400.

REFERENCES

[1] C. Shiranthika, N. Premakumara, H.-L. Chiu, H. Samani, C. Shyalika, and C.-Y. Yang, "Human Activity Recognition Using CNN & LSTM," *2020 5th International Conference on Information Technology Research (ICITR)*, pp. 1–6, Dec. 2020, doi:10.1109/icitr51448.2020.9310792.

[2] K. Hu, J. Jin, F. Zheng, L. Weng, and Y. Ding, "Overview of behavior recognition based on deep learning," *Artificial Intelligence Review*, vol. 56, no. 3, pp. 1833–1865, Jun. 2022, doi: 10.1007/s10462-022-10210-8.

[3] V. Mahalakshmi et al., "Few-shot learning-based human behavior recognition model," *Computers in Human Behavior*, vol. 151, p. 108038, Feb. 2024, doi: 10.1016/j.chb.2023.108038.

[4] A. Martin-Cirera, M. Nowak, T. Norton, U. Auer, and M. Oczak, "Comparison of Transformers with LSTM for classification of the behavioural time budget in horses based on video data," *Biosystems Engineering*, vol. 242, pp. 154–168, Jun. 2024, doi:10.1016/j.biosystemseng.2024.04.014.

[5] T. Guo, H. Liu, Z. Chen, M. Liu, T. Wang, and R. Ding, "Contrastive Learning from Extremely Augmented Skeleton Sequences for Self-Supervised Action Recognition," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 1, pp. 762–770, Jun. 2022, doi:10.1609/aaai.v36i1.19957.

[6] J. Li, H. Liu, C. Zhang, K. Li, and Y. Sun, "Interaction Recognition Using Depth Information Based on 3D CNNs," *2019 IEEE International Conference on Signal, Information and Data Processing (ICSIDP)*, pp. 1–7, Dec. 2019, doi:10.1109/icsidp47821.2019.9172824.

[7] S. Kahlouche and M. Belhocine, "Human Action Recognition using Convolutional Neural Network: Case of Service Robot Interaction," *Proceedings of the 19th International Conference on Informatics in Control, Automation and Robotics*, pp. 105–112, 2022, doi:10.5220/0011122300003271.

[8] J. Hu, B. Weng, T. Huang, J. Gao, F. Ye, and L. You, "Deep Residual Convolutional Neural Network Combining Dropout and Transfer Learning for ENSO Forecasting," *Geophysical Research Letters*, vol. 48, no. 24, Dec. 2021, doi: 10.1029/2021gl093531.

[9] Jiang Wang, Zicheng Liu, Ying Wu, and Junsong Yuan, "Mining actionlet ensemble for action recognition with depth cameras," 2012 *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1290–1297, Jun. 2012, doi: 10.1109/cvpr.2012.6247813.

[10] O. Oreifej and Z. Liu, "HON4D: Histogram of Oriented 4D Normals for Activity Recognition from Depth Sequences," *2013 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2013, doi: 10.1109/cvpr.2013.98.

[11] H. Pazhoumand-Dar, C.-P. Lam, and M. Masek, "Joint movement similarities for robust 3D action recognition using skeletal data," *Journal of Visual Communication and Image Representation*, vol. 30, pp. 10–21, Jul. 2015, doi: 10.1016/j.jvcir.2015.03.002.

[12] X. Yang and Y. Tian, "Super Normal Vector for Human Activity Recognition with Depth Cameras," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 5, pp. 1028–1039, May 2017, doi: 10.1109/tpami.2016.2565479.

[13] C. Lu, J. Jia, and C.-K. Tang, "Range-Sample Depth Feature for Action Recognition," *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 772–779, Jun. 2014, doi: 10.1109/cvpr.2014.104.

[14] Y. Du, Y. Fu, and L. Wang, "Representation Learning of Temporal Dynamics for Skeleton-Based Action Recognition," *IEEE Transactions on Image Processing*, vol. 25, no. 7, pp. 3010–3022, Jul. 2016, doi: 10.1109/tip.2016.2552404.

[15] Z. Shi and T.-K. Kim, "Learning and Refining of Privileged Information-Based RNNs for Action Recognition from Depth Sequences," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4684–4693, Jul. 2017, doi:10.1109/cvpr.2017.498.

[16] R. Parhi and R. D. Nowak, "The Role of Neural Network Activation Functions," IEEE Signal Processing Letters, vol. 27, pp. 1779–1783, 2020, doi: 10.1109/lsp.2020.3027517.

[17] K. A. Athira and J. Divya Udayan, "Temporal Fusion of Time-Distributed VGG-16 and LSTM for Precise Action Recognition in Video Sequences," *Procedia Computer Science*, vol. 233, pp. 892–901, 2024, doi: 10.1016/j.procs.2024.03.278.

[18] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, T. Darrell." Long-term recurrent convolutional networks for visual recognition and description", "*2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*," 2015, doi: 10.1109/cvpr31182.2015.

[19] S. Aburass, O. Dorgham, and J. A. Shaqsi, "A hybrid machine learning model for classifying gene mutations in cancer using LSTM, BiLSTM, CNN, GRU, and GloVe," *Systems and Soft Computing*, vol. 6, p. 200110, Dec. 2024, doi: 10.1016/j.sasc.2024.200110.

[20] H. Wang, Y. Zhang, J. Liang, and L. Liu, "DAFA-BiLSTM: Deep Autoregression Feature Augmented Bidirectional LSTM network for time series prediction," *Neural Networks*, vol. 157, pp. 240–256, Jan. 2023, doi: 10.1016/j.neunet.2022.10.009.

[21] H. Rao, S. Xu, X. Hu, J. Cheng, and B. Hu, "Augmented Skeleton Based Contrastive Action Learning with Momentum LSTM for Unsupervised Action Recognition," Information Sciences, vol. 569, pp. 90–109, Aug. 2021, doi: 10.1016/j.ins.2021.04.023.

[22] T. Zhou, A. Tao, L. Sun, B. Qu, Y. Wang, and H. Huang, "Behavior recognition based on the improved density clustering and context-guided Bi-LSTM model," *Multimedia Tools and Applications*, vol. 82, no. 29, pp. 45471–45488, May 2023, doi: 10.1007/s11042-023-15501-y.

[23] Y. Chen, Z. Zhang, C. Yuan, B. Li, Y. Deng, and W. Hu, "Channel-wise Topology Refinement Graph Convolution for Skeleton-Based Action Recognition," 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Oct. 2021, doi: 10.1109/iccv48922.2021.01311.

[24] N. Zhang, Y. Song, D. Fang, Z. Gao, and Y. Yan, "An Improved Deep Convolutional LSTM for Human Activity Recognition Using Wearable Sensors," IEEE Sensors Journal, vol. 24, no. 2, pp. 1717–1729, Jan. 2024, doi: 10.1109/jsen.2023.3335213.

[25] P. Lalwani and G. Ramasamy, "Human activity recognition using a multi-branched CNN-BiLSTM-BiGRU model," Applied Soft Computing, vol. 154, p. 111344, Mar. 2024, doi: 10.1016/j.asoc.2024.111344.

[26] C. Cheng and H. Xu, "A 3D motion image recognition model based on 3D CNN-GRU model and attention mechanism," Image and Vision Computing, vol. 146, p. 104991, Jun. 2024, doi: 10.1016/j.imavis.2024.104991.

[27] S. Fothergill, H. Mentis, P. Kohli, and S. Nowozin, “Instructing people for training gestural interactive systems,” Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, May 2012, doi: 10.1145/2207676.2208303.

[28] M. E. Hussein, M. Torki, M. A. Gowayyed and M. El-Saban, "Human action recognition using a temporal hierarchy of covariance descriptors on 3D joint locations", Proc. Int. Joint Conf. Artif. Intell., pp. 2466-2472, 2013.

[29] Y. Du, Y. Fu, and L. Wang, “Skeleton based action recognition with convolutional neural network,” 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR), pp. 579–583, Nov. 2015, doi: 10.1109/acpr.2015.7486569.

[30] P. Wang, Z. Li, Y. Hou, and W. Li, “Action Recognition Based on Joint Trajectory Maps Using Convolutional Neural Networks,” Proceedings of the 24th ACM international conference on Multimedia, Oct. 2016, doi: 10.1145/2964284.2967191.

[31] R. Ibañez, Á. Soria, A. Teyseyre, G. Rodríguez, and M. Campo, “Approximate string matching: A lightweight approach to recognize gestures with Kinect,” Pattern Recognition, vol. 62, pp. 73–86, Feb. 2017, doi: 10.1016/j.patcog.2016.08.022.

[32] F. Deboeverie, S. Roegiers, G. Allebosch, P. Veelaert, and W. Philips, “Human gesture classification by brute-force machine learning for exergaming in physiotherapy,” 2016 IEEE Conference on Computational Intelligence and Games (CIG), pp. 1–7, Sep. 2016, doi: 10.1109/cig.2016.7860414.

[33] F. Meng, H. Liu, Y. Liang, M. Liu, and W. Liu, “Hierarchical Dropped Convolutional Neural Network for Speed Insensitive Human Action Recognition,” 2018 IEEE International Conference on Multimedia and Expo (ICME), pp. 1–6, Jul. 2018, doi: 10.1109/icme.2018.8486477.