 DOI: 10.35382/TVUJS.14.8.2024.84

BEHAVIOR RECOGNITION WITH LSTM DEEP LEARNING MODEL AND MEDIAPIPE

Tran Song Toan[1], Minh-Hai Le[2*], Huu-Phuc Dang[3]

Abstract – *Human behavior recognition is crucial for assisting and monitoring the activities of patients, particularly, the elderly or young children. With the advancement of technology, modern methods based on computer vision have been developing. Deep learning is one of the prominent methods for dealing with problems related to behavior recognition. In this study, a long short-term memory deep learning model is used for identifying abnormal behaviors. The MediaPipe library is used to collect body points and consecutive frames to generate training data and recognition. The behaviors considered in this paper include headache, stomachache, fall down, and others. With the dataset collected from videos and self-recorded, the experimental results show that the proposed long short-term memory network model achieves 94.54% accuracy in behavior recognition. This result demonstrates the feasibility of the proposed model for the task of behavior recognition.*

Keywords: deep learning, human behavior recognition, long short-term memory (LSTM), MediaPipe.

I. INTRODUCTION

Human action recognition (HAR) is an interdisciplinary field that combines computer vision with other disciplines to analyze human movement, balance, posture control, and interaction with the environment. It encompasses areas such as biomechanics, computer vision, image processing, data analysis, nonlinear modeling, artificial intelligence, and pattern recognition. HAR can be analyzed using two-dimensional, depth, or thermal images or motion, body-mounted sensors, or smartphones. HAR has been extensively researched due to its numerous applications in various domains and complexities, with prominent applications in safety, environmental monitoring, video surveillance [1, 2], robotics [3], and the like. In the context of safety and video surveillance, HAR can be employed to assist in monitoring the elderly, patients, or children.

Another approach to behavior recognition involves analyzing captured images. Employing image processing algorithms and deep learning models, one can extract meaningful insights into the actions of individuals within videos. Deep learning methods have gained prominence in this domain due to their exceptional generalization capabilities and the ability to circumvent the need for handcrafted feature extraction [4].

Convolutional neural networks (CNNs) have emerged as a popular choice for behavior recognition tasks [5]. Long short-term memory (LSTM) networks have also gained significant traction in this domain [6]. The effectiveness of deep learning models for behavior recognition hinges on the ability to extract relevant features that facilitate accurate and efficient classification. In the context of HAR, the features of interest are the positions of keypoints on the human body across a sequence of video frames. CNNs, while powerful for feature extraction, may not be well-suited for directly capturing these keypoints. To address this limitation, the proposed approach utilizes the MediaPipe library [7] to extract keypoints and combines features from consecutive frames as input to a LSTM network. This model can be implemented without demanding hardware requirements. The dataset employed in this study comprises videos collected from YouTube and self-recorded videos.

[1,2,3]Tra Vinh University, Vietnam
*Corresponding author: lmhai@tvu.edu.vn
Received date: 09th July 2024; Revised date: 13th September 2024; Accepted date: 19th September 2024

The presented study makes significant contributions to the field of HAR, particularly in the context of video-based behavior analysis. The key contributions are summarized as follows:

1. Leveraging MediaPipe for feature extraction: The proposed approach utilizes the MediaPipe library to extract keypoints from video frames, providing a robust and efficient mechanism for feature extraction.

2. Combining MediaPipe with LSTM and neural networks: The study effectively integrates MediaPipe-extracted keypoints with an LSTM network and neural network architecture for behavior recognition. This combination harnesses the strengths of both techniques to achieve accurate and efficient classification.

3. Comprehensive dataset evaluation: The research employs a comprehensive dataset comprised of both publicly available and self-collected videos to evaluate the proposed model. This diverse dataset ensures the robustness and generalizability of the findings.

II. METHODOLOGY

A. System overview

This workflow highlights the key steps involved in the proposed human behavior recognition system. The system effectively utilizes MediaPipe pose for feature extraction and an LSTM model for behavior classification, demonstrating a promising approach for analyzing and interpreting human actions in videos.

The proposed system for human behavior recognition follows the workflow depicted in Figure 1. The system receives a video as input. MediaPipe pose is employed to extract 33 skeletal points from each frame of the video. Each skeletal point is represented by four coordinates: (x, y, z, visibility). These coordinates are flattened into a 132-dimensional array (33 points * 4 coordinates). Consecutive 30 frames of the flattened coordinate arrays are combined into a sequence. This sequence serves as the input to the LSTM model. The trained LSTM model analyzes the input sequence. The model classifies the sequence into one of the predefined behavior categories. The system outputs the recognized behavior as the final result.

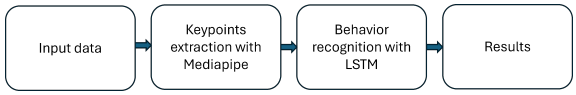


Fig. 1: System overview

The study focuses on recognizing three primary behaviors: falling, headache, and stomachache. Any other actions are classified as ‘other behavior’. Figure 2 illustrates the data collection process for training the LSTM model. Videos with corresponding behaviors are collected. Each video has a duration of 10 minutes with a resolution of 720 x 1280 pixels. Different body parts are identified in each frame. The movement of these body parts is analyzed over time. Upon pose detection, 33 body keypoints are extracted.

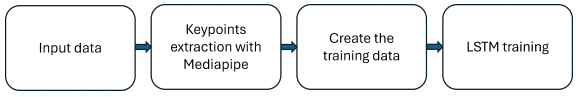


Fig. 2: Data collection and sampling for training

B. LSTM model

LSTM networks are a type of recurrent neural network (RNN) specifically designed to handle sequential data with long-term dependencies. They have emerged as powerful tools for processing and understanding time-series data, particularly in scenarios where long-range patterns and dependencies are crucial. LSTM networks incorporate memory cells that enable them to store and retain information over extended periods, addressing the vanishing gradient problem that plagues traditional RNNs. LSTM networks excel at capturing long-term dependencies in sequential data, making them suitable for tasks like natural language processing, speech recognition, and time-series forecasting.

LSTM networks utilize three main gates: the input gate, the forget gate, and the output gate. These gates regulate the flow of information into,

through, and out of the memory cells, allowing for selective memory updates and controlled information processing.

Forget Gate – f_t: The forget gate determines what portion of the current cell state is retained, selectively erasing outdated or irrelevant information.

$$f_t = \sigma(U_f * x_t + W_f * h_{t-1} + b_f) \quad (1)$$

Input Gate – i_t: The input gate controls the extent to which new information is added to the cell state.

$$i_t = \sigma(U_i * x_t + W_f * h_{t-1} + b_i) \quad (2)$$

Output Gate – o_t: The output gate regulates the flow of information from the cell state to the output of the block, determining what information is made available to subsequent blocks.

$$o_t = \sigma(U_o * x_t + W_o * h_{t-1} + b_o) \quad (3)$$

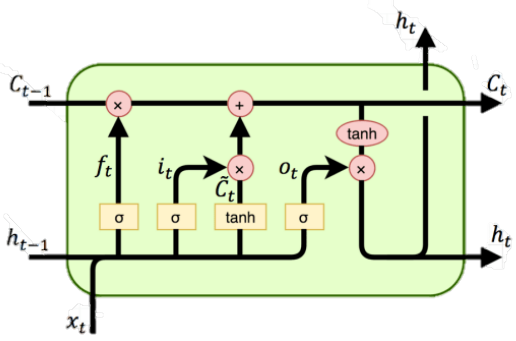


Fig. 3: The state of an LSTM (long short-term memory) network at time step t [8]

Where c_t is the cell state, h_t denotes the hidden state, x_t is the t^{th} input of the model and h_{t-1}, c_{t-1} presents the output of the previous layer.

The LSTM model employed in the study consists of eight layers, as illustrated in Figure 4. The core of the model comprises four LSTM layers with varying numbers of neurons: [64, 128, 256, 256]. These layers capture and process temporal dependencies in the sequential data. Following the LSTM layers, a layer normalization layer is introduced. This layer normalizes the activation values of the previous layer, improving the training process and overall model performance. Four fully connected layers are used to combine and transform extracted features for classification. The softmax activation layer is used to generate a probability distribution over the four behavior classes (falling, headache, stomachache, and others). The specific training parameters and hyperparameters for the LSTM model are presented in Table 1. These parameters govern the optimization process and influence the model's learning behavior.

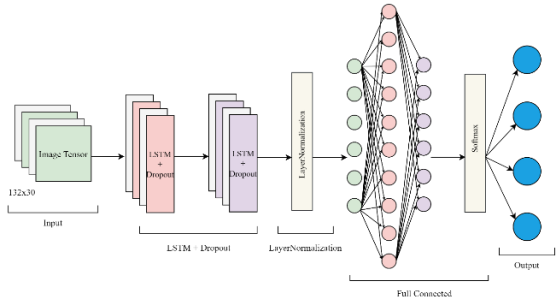


Fig. 4: The LSTM model used in this study

Table 1: The parameters of the model

Parameter	Value
LSTM hidden layer	[64, 128, 256, 256]
Dropout rate	0.2
Layer Normalization	1
Dense neural node	[128, 256, 64, 4]

C. Mediapipe pose

MediaPipe pose is an open-source real-time human pose estimation solution developed by Google. It utilizes machine learning models to estimate the 3D positions of 33 keypoints on the human body from videos or images. These keypoints provide valuable information about an individual's posture and movements.

In this study, the 33 keypoints extracted from MediaPipe pose are collected continuously over

30 frames, forming a data sample for training the LSTM model. The model utilizes this data to recognize human behaviors. Each of the 33 keypoints has four corresponding values: x, y, z, and visibility. These values represent the 3D coordinates (x, y, z) of the keypoint and its visibility score (0 for invisible, 1 for fully visible). To prepare the data for the LSTM model, an array was created by concatenating the four corresponding values for each keypoint. This results in a 132-dimensional array (33 keypoints * 4 values per keypoint). This array serves as the input to the LSTM model.

The integration of MediaPipe pose with the LSTM model enables the system to effectively process sequential pose data and recognize human behaviors from videos or images. MediaPipe pose provides accurate 3D pose estimation, while the LSTM model excels at handling temporal dependencies and classifying behaviors based on the pose sequences.

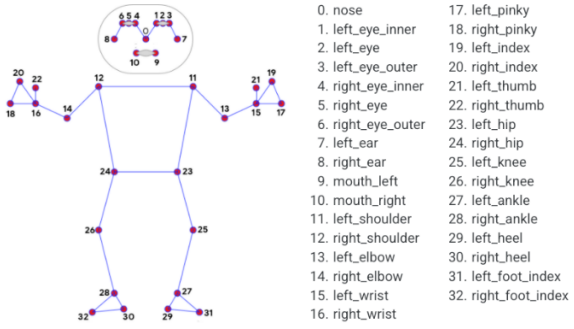


Fig. 5: The 33 keypoints of MediaPipe pose

III. EXPERIMENTS

A. Datasets

A dataset was constructed for model training and evaluation through the collection and analysis of video data. Data sources included social media platforms and original recordings. Subsequent to data acquisition, a meticulous analysis and labeling process was undertaken according to the methodology outlined in Section II-A. A comprehensive overview of the dataset employed in this study is presented in Table 2.

To effectively evaluate the model's performance, the initial dataset was divided into training and evaluation sets. The training set is used to train the model, while the evaluation set is used to assess its performance. The training set comprises 80% of the initial dataset, while the evaluation set comprises 20%. The detailed sample sizes are presented in Table 2.

Table 2: Amount of data for experimental

Behaviors	Training	Testing	Total
Headache	15200	3800	19000
Fall down	15200	3800	19000
Stomach-ache	15200	3800	19000
Others	15200	3800	19000
Total	60800	15200	76000

B. Experimental setting

The training process was carried out on a personal computer equipped with a Ryzen 5000 CPU, an RTX1050 GPU, and 8GB of RAM, using the PyCharm software. The training parameters are detailed in Table 3.

Table 3: Hyperparameter of the model

Hyperparameter	Value
Learning rate	0.01
Batch_size	32
Epochs	100
Frames	30
Optimizer	Adam optimizer
Loss function	Categorical Cross Entropy

Categorical cross-entropy is a commonly used loss function for measuring the difference between true and predicted probability distributions in multi-class classification tasks. The formula for categorical cross-entropy for a single data sample is presented in Expression (4).

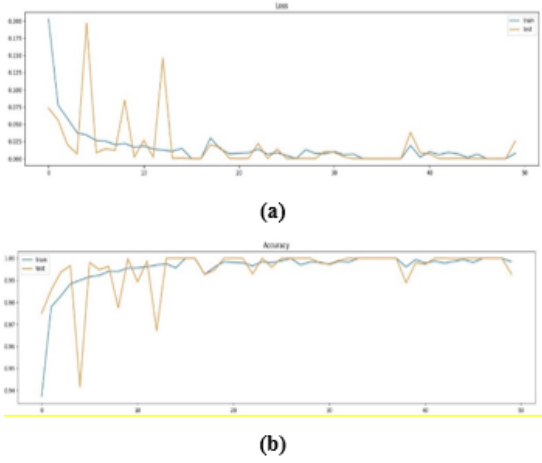
$$H(y,\hat{y}) = -\sum_i^n y_i \log(\hat{y}_i) \qquad (4)$$

where $H(y,\hat{y})$ is the categorical cross-entropy loss function, n denotes the class number, y_i is the ground truth and $\hat{y}_i$ show the predicted class i.

C. Traing process

The model was trained for 100 epochs and then evaluated using loss and accuracy metrics. The model employed an early stop callback to halt the training process prematurely if the monitored parameter (either validation accuracy or validation loss) did not improve after a specified number of epochs. The training results indicate that the model achieved satisfactory performance after 50 epochs and was terminated due to the early stop mechanism.

Figures 6(a) and 6(b) depict the loss and accuracy curves over 50 epochs, with the blue line representing the training dataset values and the orange line representing the test dataset values. The loss and accuracy values exhibit an inverse relationship. During the initial 15 epochs, the values are unstable, the accuracy is low, and the test data is not yet very accurate. From approximately epochs 15 to 50, the values gradually stabilize, become more accurate, and the difference between the training and test data diminishes. Figure 7 demonstrates that the model's accuracy on the test set even surpasses that of the training set, indicating that the model is performing quite well. As the number of epochs increases, both train loss and test loss decrease, while train accuracy and test accuracy increase.



(a)

(b)

Fig. 6: The learning curves: (a) Loss value and (b) Accuracy value

IV. RESULTS AND DISCUSSION

A. The metrics

Accuracy, Precision, Recall, and F1-score are metrics employed to evaluate the training and testing processes of test samples. The formulas for determining Accuracy, Precision, Recall, and F1-score are presented in Expressions (5), (6), (7), and (8).

$$Accuracy = \frac{TP+TN}{N} \quad (5)$$

$$Precision = \frac{TP}{TP+FP} \quad (6)$$

$$Recall = \frac{TP}{TP+FN} \quad (7)$$

$$F1-Score = \frac{TP}{TP+\frac{1}{2}(FP+FN)} \quad (8)$$

where TP (true positive): the total number of cases where the model correctly predicted a positive sample; TN (true negative): the total number of cases where the model correctly predicted a negative sample; FP (false positive): the total number of cases where the model incorrectly predicted a negative sample as positive; FN (false negative): the total number of cases where the model incorrectly predicted a positive sample as negative; N: the total number of samples used for prediction.

B. Experimental results

Utilizing the sample sizes of the training and testing sets, the sklearn library was employed to compute the model's metrics. The model's performance is summarized in Table 4.

Table 4: LSTM model training results

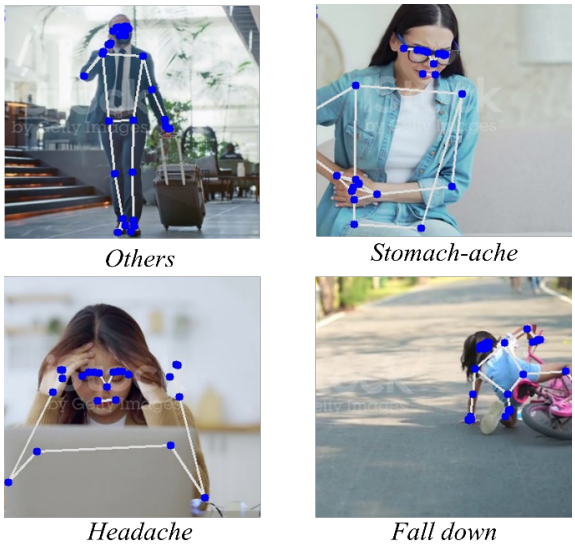
Metrics	Acccuracy	Precision	Recall	F1 Score
Results (%)	94.54	96.66	94.57	94.22

To evaluate the overall system's performance, 20 real behavioral samples were tested. The results presented in Table 5 indicate that the overall system identification rate is lower due to the dependency on the keypoint collection process. Since the 'other behavior' in the synthetic dataset used for pretraining involves simpler movements

compared to those in the real dataset, the prediction results for identifying 'other behavior' on the real dataset frequently misclassify it as 'Stomach-ache' or 'Fall down'. 'Stomach-ache' can be easily confused with normal behavior as these two actions are quite similar, except that in 'Stomach-ache' the user will place their hand on their abdomen. For 'Fall down' the model performs better than for other behaviors. Figure 7 presents some correct behavior identification results from the frames.

Table 5: Confusion matrix in the real set with 20 samples

	Stomach-ache	Headache	Fall down	Others
Stomach-ache	12	2	2	4
Headache	2	13	3	2
Fall down	2	0	16	2
Others	4	1	4	11



Others *Stomach-ache*

Headache *Fall down*

Fig. 7: Behavior recognition results

V. CONCLUSION

The paper presents a novel approach for identifying patient behaviors using a deep learning LSTM model. The system utilizes the MediaPipe library to extract keypoints from human postures and constructs a recognition dataset by combining consecutive frames. The proposed model demonstrates promising performance through training and testing evaluations. While some inaccuracies remain in real-world system operation, the model showcases the potential and feasibility of human behavior recognition from videos.

REFERENCES

[1] Babiker M, Khalifa OO, Htike KK, Hassan A, Zaharadeen M. Automated daily human activity recognition for video surveillance using neural network. In: *2017 IEEE 4th international Conference on Smart Instrumentation, Measurement and Application (ICSIMA)*. 28th–30th November 2017; Putrajaya, Malaysia. IEEE; 2017. p.1–5. https://doi.org/10.1109/ICSIMA.2017.8312024.

[2] Taha A, Zayed H, Khalifa ME, El-Horbaty ME. A human activity recognition for surveillance applications. In: *Proceedings of the 7th International Conference on Information Technology (ICIT)*. Al-Zaytoonah University of Jordan; 2015. p.577–586. https://doi.org/10.15849/icit.2015.0103.

[3] Piyathilaka L, Kodagoda S. Human activity recognition for domestic robots. In: *Field and Service Robotics: Results of the 9th International Conference*. Berlin/Heidelberg, Germany: Springer; 2015. p.395–408. https://doi.org/10.1007/978-3-319-07488-7_27.

[4] Rodriguez-Moreno I, Martínez-Otzeta JM, Sierra B, Rodriguez I, Jauregi E. Video activity recognition: State-of-the-art. *Sensors*. 2019;19(14):3160. https://doi.org/10.3390/s19143160.

[5] Cherian A, Fernando B, Harandi M, Gould S. Generalized rank pooling for activity recognition. In: *The 2017 Conference on Computer Vision and Pattern Recognition*. Honolulu, HI, USA: IEEE; 2017. p.1631–1640.

[6] Dai C, Liu X, Lai J. Human action recognition using two-stream attention based LSTM networks. *Applied Soft Computing*. 2020;86: 105820. https://doi.org/10.1016/j.asoc.2019.105820.

[7] Camillo L, Jiuqiang T, Hadon N, Chris MC, Esha U, Michael H, et al. MediaPipe: A framework for building perception pipelines. *ArXiv* [Preprint] 2019. https://doi.org/10.48550/arXiv.1906.08172.

[8] Zhou T, Ruan S, Canu S. A review: Deep learning for medical image segmentation using multi-modality fusion. *Array*. 2019;3–4: 100004. https://doi.org/10.1016/j.array.2019.100004.