International Journal of Advanced Science and Technology
Vol. 29, No. 7s, (2020), pp. 3237-3244

3D Skeleton-based Action Recognition Using LSTM Network

Mega Cynthia Wishnu[1], Lukas[2], Duma Kristina Yanti Hutapea[3] and Wen-Nung Lie[4]

[1]Department of Electrical Engineering, Faculty of Engineering, Atma Jaya Catholic University of Indonesia, Jakarta 12930, Indonesia
[2,3]Cognitive Engineering Research Group (CERG), Faculty of Engineering, Atma Jaya Catholic University of Indonesia, Jakarta 12930, Indonesia
[4]Department of Electrical Engineering, National Chung Cheng University, Chia-Yi, Taiwan, ROC
[1]mega.201800090002@student.atmajaya.ac.id, [2]lukas@atmajaya.ac.id, [3]duma.kristina@atmajaya.ac.id, [4]ieewnl@ccu.edu.tw

Abstract

Human action recognition is an interesting Artificial Intelligence research in recognizing human action pattern. It is very usefull in many application such such as video surveillance, human-machine interaction and video analysis. One of data modalities that can be used is 3D skeleton data that extracted from depth sensor. Deep learning method is picked to solve this problem. Furthermore, Long-Short Term Memory (LSTM) network is applied to human action recognition problem. The methods are divided into two parts: data pre-processing and network modelling. Data pre-processing is applied so that the data can be fed to neural network. Then, the network modelling that are used in this experiment are: 1-layer LSTM and second network is 2-layer LSTM. Both networks are following by a Fully Connected layer and a softmax classifier to predict the action label. This network is evaluated on NTU RGB-D 120 dataset with two evaluation benchmark: cross-setup and cross-subject. Based on the experiment, the second network has better result than the first network. Cross-setup training accuracy is 43.64% and testing accuracy is 13.69%. Cross-subject training accuracy is 46.97% and testing accuracy is 26.02%.

Keywords: *3D skeleton, deep learning, human action recognition, LSTM*

1. Introduction

Human action recognition is an interesting Artificial Intelligence research in recognizing human action pattern. It is very usefull in many application such such as video surveillance, human-machine interaction and video analysis. By detecting human action, we can predict further decision or prevent further effects. Skeletal tracking can be obtain using depth sensors. Nowdays many depth sensors are being developed, such as Microsoft Kinect and Intel Realsense. The output of depth sensors are many kinds: Reg-Green-Blue (RGB) videos, Infrared videos, depth maps and 3D skeleton data. These modalities can be chosen as the input of human action recognition process. In this paper, 3D skeleton data is used to detect human action because it require much less computation than the other data modalities.

Deep learning method is widely used to solve human action recognition problem. Videos can be processed one at the time or extracted first into frames before being fed into the neural network models. One of deep learning method that often applied into human action recognition problem is Long-Short Term Memory (LSTM). LSTM can learn previous frame information and then compare it to current frame information. For videos data, LSTM is a perfect solution because data can be fed directly [1-3]

This network is evaluated on NTU RGB-D 120 dataset which is a large-scale benchmark dataset for 3D human activity analysis. It consists of 114.480 RGB+D video samples and has 106 different subjects. The two evaluation benchmarks for this dataset are cross-setup and cross-subject [4].

Cross-setup is an evaluation benchmark by its camera setup in [4]. There are 32 camera setups with different height and distance combination. Evaluating by cross-setup means dividing setups into training and testing data, so each group has 16 camera setups. While cross-subject evaluation is an evaluation benchmark by its subject or person. There are 106 different person recorded in [4], so each training and testing has 53 subjects. The IDs of subject can be seen in [4].

2. Methods

The methods are divided into two parts: data pre-processing and network modelling.

2.1. Data Pre-processing

The dataset used in this experiment is NTU RGB-D 120 dataset. The dataset is collected by Microsoft Kinect sensors and has four major data modalities, which are Red-Green-Blue (RGB) frames, Infrared (IR) frames, depth maps and the 3D joint information (skeleton data) [4]. Each data modality example can be seen in Figure 1 to Figure 3. The data modalities need in this experiment is only 3D joint information (skeleton data). Skeleton data consist of 3-dimensional coordinates of each joint that detected in human body from head to toe. So the total data is 75 coordinates. The 25 joints illustration can be seen in Figure 4.

NTU RGB-D dataset has 120 action categories. There are 106 distinct subjects from 15 different countries, ages are between 10 and 57 and heights are between 1.3 m and 1.9 m. Furthermore, it also has 32 collection camera setups (height and distance) with different location and background. There are three cameras used in this dataset with different horizontal angles: 45^o, 0^o, -45^o [4].

The skeleton datasets are divided into two categories: cross-setup and cross-subject. Cross-setup divides the data by its camera setting (different height and distance). There are 32 setups used for cross-setup validation. Cross-subject divides the data by its subject or performer. There are 106 subjects used for cross-setup validation [4].

The pre-processing consists of two steps. First step is convert all raw skeleton data to mat files and save only 3D coordinates (x, y, z) information using Matlab. Every skeleton file captured by Microsoft Kinect consists of 11 informations which are camera coordinates (x, y, z), color coordinates (x, y), depth coordinates (x, y), and orientation coordinates (x, y, z, w). Camera coordinates are use for joint positioning in 3D projects, color coordinates are the coordinates of the joint on the image from the color (RGB) camera, depth coordinates are the coordinates of the joint on the image from the depth camera. In this experiment, the camera coordinates (x, y, z) are used which also known as 3D coordinates of 25 joints. Using Matlab program, the other data are removed and save only the 3D coordinates data.

Second step is convert all mat files to numpy array to concatenate all joints and frames information using Python. The x, y, z coordinate from 25 joints are concatenate to one row, so each row consists of 75 data. While rows represent joint data, coloumns represent the number of frames. The size of array will be changed to 75 x frames. As mentioned above, to evaluate the dataset, there are two benchmark: cross-setup and cross-subject. Therefore, there are 8 numpy data as following: input training, output training, input testing, output testing for each cross-setup and cross-subject benchmark. Overall data pre-processing step can be seen in Figure 5.

2.2. Network Modelling

LSTM is employed to construct the network. LSTM network is a type of Recurrent Neural Network (RNN), which is a special type of neural network designed for sequence problems. LSTM is usually used for time-series problems or sequencing problems. LSTM compares previous information with current information. It is perfect for video recognition because video consists of sequential frame. The schema of LSTM unit can be seen in Figure 6.

In this experiment, two network architectures are applied to compare the validation process. First network is 1-layer LSTM following by a Fully Connected layer and a softmax classifier to predict the action label. Second network is 2-layer LSTM following by a Fully Connected layer and a softmax classifier to predict the action label. The network architecture can be seen in Figure 7 and Figure 8.

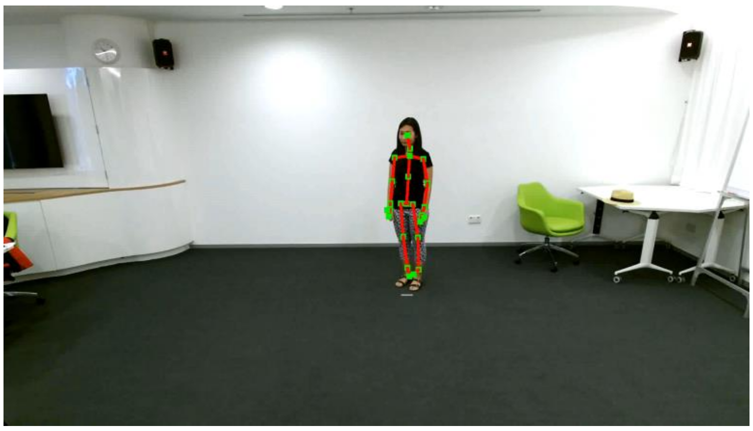


Figure 1. RGB + Skeleton Data

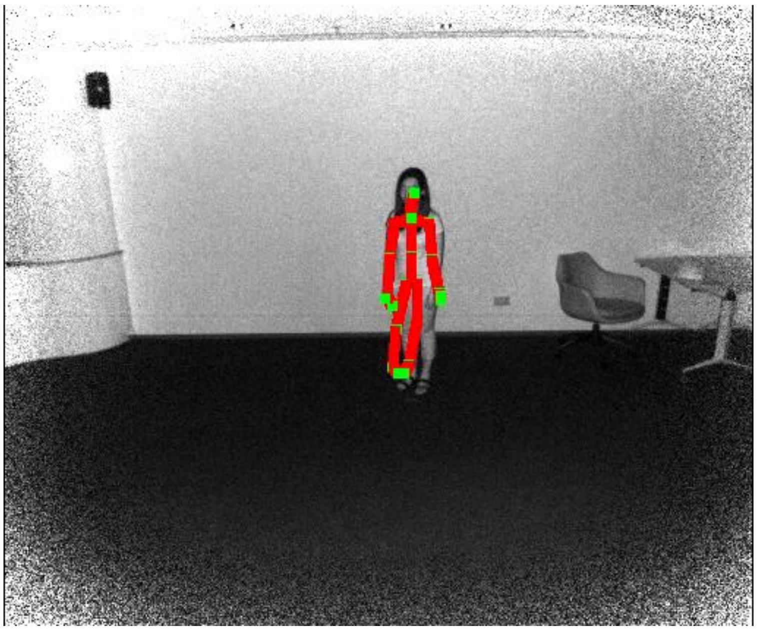


Figure 2. IR + Skeleton Data

ISSN: 2005-4238 IJAST

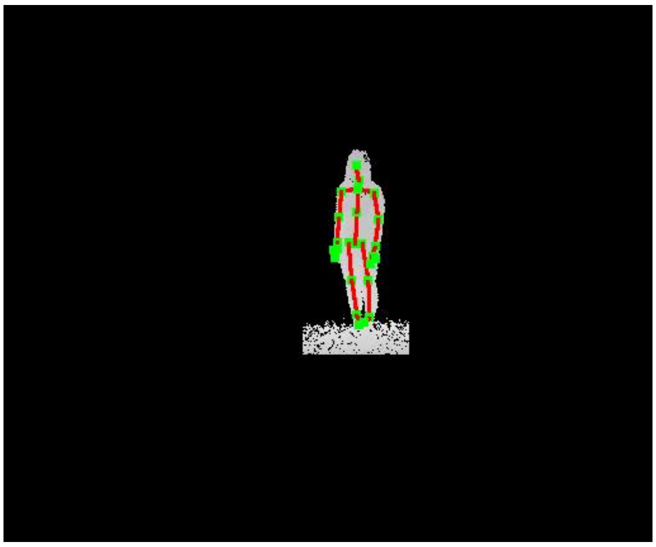


Figure 3. Depth Maps + Skeleton Data

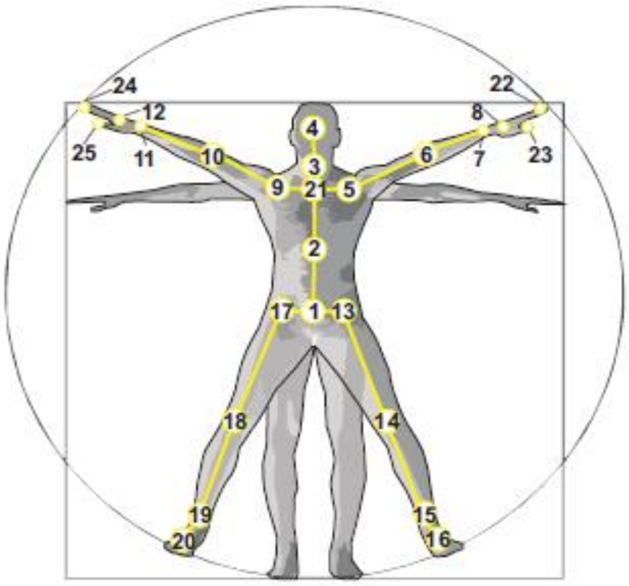


Figure 4. 25 Joints Illustration

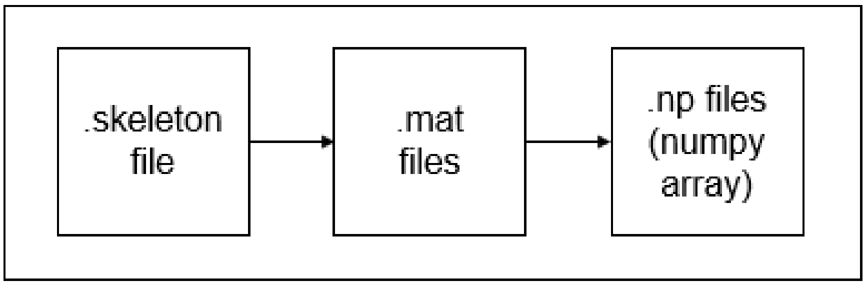


Figure 5. Data Pre-processing Steps

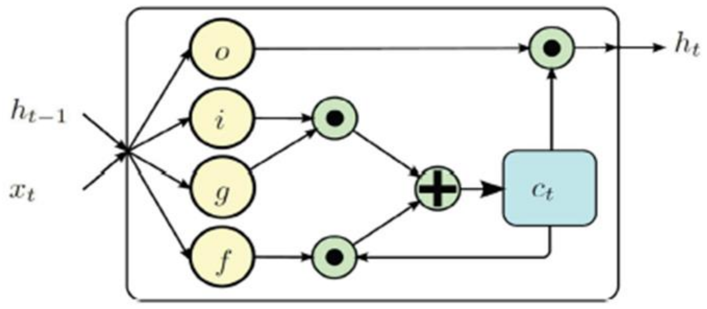


Figure 6. The Schema of LSTM unit [5]

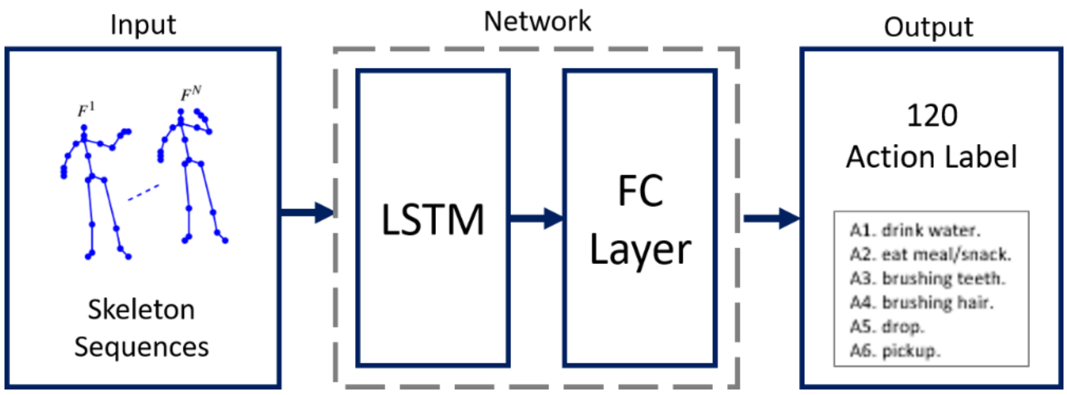


Figure 7. 1-layer LSTM network architecture

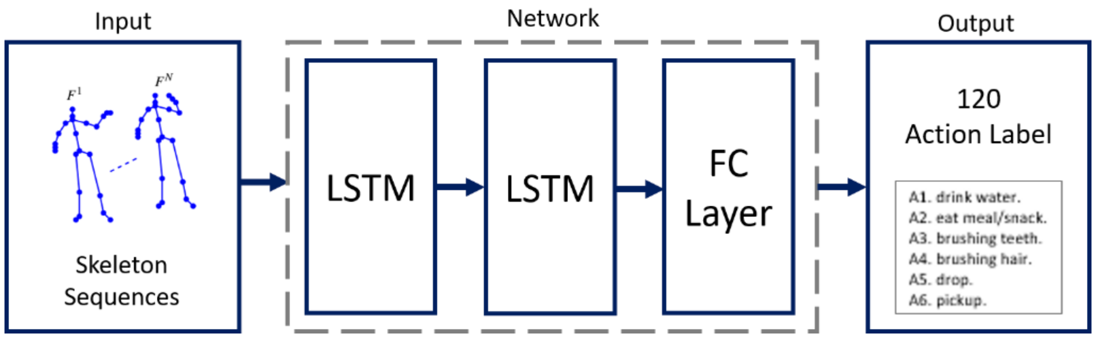


Figure 8. 2-layer LSTM network architecture

The difference of both network architectures is only in number of LSTM layer while the others remain the same. The hidden unit of LSTM layer is set to 256, the hidden unit of fully connected network is set to 128 and the last is a softmax classifier with 120 action label. Every layer has a dropout layer with rate 0.5. The network has the categorical crossentropy loss to calculate the loss, a adam optimizer with learning rate 0.0005 and weight decay 0.00001. The network is training for 50 epoch. The network is implement by tensorflow on a workstation with an NVIDIA GeForce GTX 1080 Ti GPU. Every epoch takes average 60 minutes to train.

3. Result and discussion

Training and testing accuracy from experiment can be seen in table 1 for 1-layer LSTM network architecture and table 2 for 2-layer LSTM network architecture. The graph from both table can be seen in Figure 8 and Figure 9.

The accuracy result was recorded at epoch 10, 20, 30, 40, 50. In the training phase, the greater the epoch, the higher the accuracy. While in the testing phase, accuracy does not indicate a significant increase. For cross-setup 1-layer LSTM, the best training accuracy is obtained at 50th epoch which is 45.15% while testing accuracy is around 11%. For cross-subject 1-layer LSTM, the best training accuracy is obtained at 50th epoch which is 38.69% while testing accuracy is around 23%. For cross-setup 2-layer LSTM, the best training accuracy is obtained at 50th epoch which is 43.64% while testing accuracy is around 13%. For cross-subject 2-layer LSTM, the best training accuracy is obtained at 50th epoch which is 46.97% while testing accuracy is around 26%. Among four results, cross-subject 2-layer LSTM shows the best result.

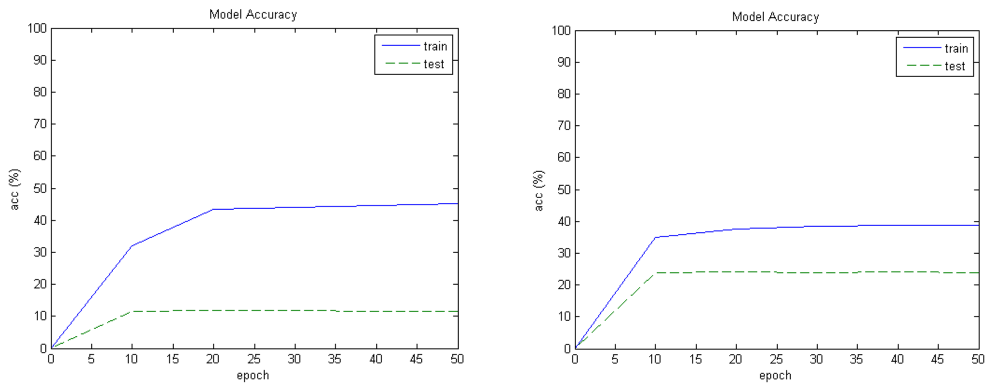
The training and testing accuracy for both networks are still very low (less than 50%). It can caused by many factors, for example because NTU RGB-D dataset is a very large dataset (up to 9 millions frames) and has too many noisy skeletons or maybe there is something wrong with our pre-processing data. Tuning the parameters in neural network are also done, for example changing the hidden unit, learning rate, weight decay, dropout regularizer, epoch and so on.

Table 1. 1-layer LSTM accuracy result

Epoch	Cross-Setup		Cross-Subject	
	Train Acc.	Test Acc.	Train Acc.	Test Acc.
10	31.92%	11.56%	35.04%	23.81%
20	43.29%	11.76%	37.52%	24.10%
30	44.02%	11.70%	38.52%	23.76%
40	44.61%	11.61%	38.61%	24.01%
50	45.15%	11.53%	38.69%	23.79%

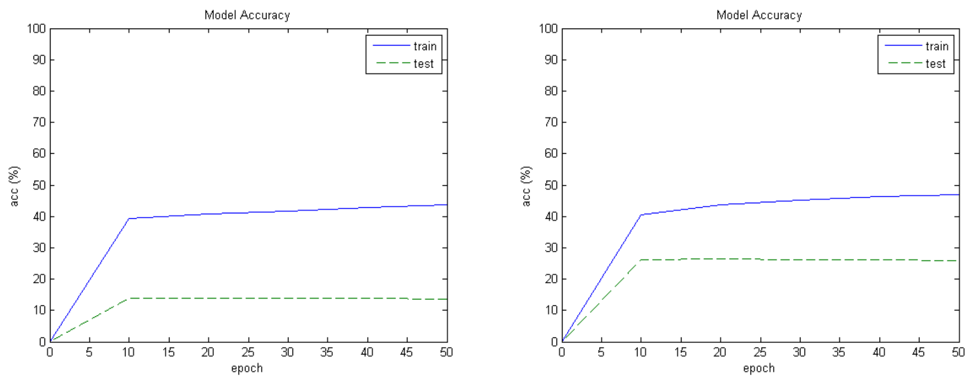
Table 2. 2-layer LSTM accuracy result

Epoch	Cross-Setup		Cross-Subject	
	Train Acc.	Test Acc.	Train Acc.	Test Acc.
10	39.24%	13.99%	40.42%	26.19%
20	40.65%	13.86%	43.67%	26.38%
30	41.81%	13.86%	45.24%	26.16%
40	42.77%	13.76%	46.25%	26.08%
50	43.64%	13.69%	46.97%	26.02%



(a) (b)

Figure 8. (a) Cross-setup 1-layer LSTM model accuracy (b) cross-subject 1-layer LSTM model accuracy



(a) (b)

Figure 9. (a) Cross-setup 2-layer LSTM model accuracy (b) cross-subject 2-layer LSTM model accuracy

4. Conclusion

As we can see from the experiment result with 1-layer LSTM, cross-setup training accuracy is 45.15% and testing accuracy is 11.53%. While for cross-subject, the training accuracy is 38.69% and testing accuracy is 23.79%. For 2-layer LSTM, cross-setup training accuracy is 43.64% and testing accuracy is 13.69%. While for cross-subject, the training accuracy is 46.97% and testing accuracy is 26.02%

For cross-setup, training accuracy in 1-layer LSTM network is higher than training accuracy in 2-layer LSTM but testing accuracy in 1-layer LSTM network is lower than testing accuracy in 2-layer LSTM. Since testing accuracy in 2-layer LSTM network is higher than in 1-layer LSTM network, it can be concluded that 2-layer LSTM network has better result than 1-layer LSTM network.

The training accuracy for both networks are lower than 50% and testing accuracy are lower than 30%. Training accuracies are increasing at each epoch while the testing accuracies are sometimes increasing and decreasing. To increase training and testing accuracy, further experiment is needed, for example by adding more epoch and data augmenting to have more various dataset.

Acknowledgments

This research was supported by Department of Electrical Engineering, Faculty of Engineering, Atma Jaya Catholic University of Indonesia collaborating with College of Engineering, National Chung Cheng University, Taiwan, Republic of China.

References

[1] Tu J, Liu H, Meng F, Liu M and Ding R 2018 Spatial-Temporal Data Augmentation Based on LSTM Autoencoder Network for Skeleton-Based Human Action Recognition (Athens: International Conference on Image Processing (ICIP) IEEE)

[2] Hoang V N, Le T L, Hai-Vu and Nguyen V T 2019 3D Skeleton-Based Action Recognition with Convolutional Neural Networks (Ho Chi Minh: International Conference on Multimedia Analysis and Pattern Recognition (MAPR))

[3] Huynh-The T, Hua C H and Kim D S 2019 Learning Action Images Using Deep Convolutional Neural Networks for 3D Action Recognition (Sophia Antipolis: Sensors Applications Symposium (SAS) IEEE)

[4] Liu J, Shahroudy A, Perez M, Wang G, Duan L Y and Kot A C 2019 NTU RGB+D 120: A Large-Scale Benchmark for 3D Human Activity Understanding (Transactions on Pattern Analysis and Machine Intelligence (TPAMI) IEEE)

[5] Li C, Hou Y, Wang P and Li W 2019 Multiview-Based 3-D Action Recognition Using Deep Networks (Bari: Transactions On Human-Machine Systems IEEE)