Markerless Human Activity Recognition Method Based on Deep Neural Network Model Using Multiple Cameras

Conference Paper · April 2018

DOI: 10.1109/CoDIT.2018.8394780

3 authors:

Prasetia Utama
Yokohama National University
2 PUBLICATIONS **2** CITATIONS
SEE PROFILE

Keisuke Shima
Yokohama National University
63 PUBLICATIONS **151** CITATIONS
SEE PROFILE

Koji Shimatani
Prefectural University of Hiroshima
58 PUBLICATIONS **65** CITATIONS
SEE PROFILE

Markerless Human Activity Recognition Method Based on Deep Neural Network Model Using Multiple Cameras

Prasetia Utama Putra
Graduate School of Engineering,
Yokohama National University
Email: prasetia-putra-kt@ynu.jp

Keisuke Shima
Faculty of Engineering,
Yokohama National University
Email: shima@ynu.ac.jp

Koji Shimatani
Faculty of Health and Welfare,
Prefectural University of Hiroshima
Email: shimatani@pu-hiroshima.ac.jp

***Abstract*—Most methods of multi-view human activity recognition can be classified as conventional computer vision approaches. Those approaches separate feature descriptor and discriminator. Hence, the feature extractor cannot learn from the mistakes made by the classifier. In this paper, a deep neural network (DNN) model for human activity estimation using multi-view sequences of raw images is presented. This approach incorporates features extractor and discriminator into a single model. The model comprises three parts, a convolutional neural network (CNN) block, MSLSTMRes, and a dense layer. This method enables discrimination of human activity such as "walk" and "sit down" by merely using sequences of raw images. Experimental results on IXMAS dataset using one-subject cross validation demonstrates high prediction rate that is comparable to other methods in the literature, which utilized preprocessed images such as silhouette and volumetric data and sophisticated feature extractor.**

I. Introduction

Human activity recognition is one of challenging issues in pattern recognition field. Its application ranges from human-computer interaction, intelligent surveillance system, to understanding human behavior. There are many attempts that have been done by researchers to tackle the issue of vision-based human activity prediction, yet they still suffer several limitations. There are some reasons that cause those limitations. First, when estimating human action an algorithm must consider deformation both in spatial and temporal space. The second is the noise produced by the environment or human itself e.g. illumination change, self-occlusion, and mutual occlusion. Third, the physical configuration of a human body that differs and is also similar from one action to the others.

In [1]–[5], they attempted to solve self-occlusion problem by utilizing information from multiple cameras. By combining information from different viewpoints to better understand about the deformation in spatial-temporal spaces; those techniques yielded better accuracy performance compared to approaches that utilized information from only one camera. Typically, existing methods for multiple-view human action recognition can be classified as conventional computer vision approaches, in which the features fed to the classifier are hand-crafted. In general, the hand-crafted features extractor technique works independently from the classifier. They do not consider the mistakes made by the classifier to produce more informative features that the classifier can better accommodate.

Different from conventional computer vision technique, deep neural network (DNN) combines feature extractor and discriminator in a single pipeline. In DNN, feature extractor and classifier are trained simultaneously. The error classification made by the classifier is propagated to the features extractor to adjust its parameters. Hence it can produce features that fit in with the discriminator. Although DNN has been widely used in vision task e.g. object recognition, automatic video labeling, and single-view activity recognition, its application in multi-view human action prediction is still in small number.

In this paper, we propose a DNN model to estimate human activity recorded from multiple cameras. The model consists of three parts, a convolutional neural network (CNN) [6] block to extract spatial information, multiple stacked long short-term memory residual (MSLSTMRes) as a new LSTM topology, to decode temporal information, and a dense layer with softmax function [6] to classify the features from all views. The performance of the model was evaluated on IXMAS dataset and was compared to previous methods using the same dataset with similar experiment protocol.

The main novel contributions of this paper are: (i) the proposal of a DNN model for multiple-view human activity recognition with raw images as the inputs, and (ii) the investigation of the effect of ensemble structure and residual learning in LSTM.

The rest of paper is organized as follow. In section II, we briefly review related methods. Section III describes the detail of our proposed model. While experiments and their results are explained in section IV and V. Finally, conclusions and possible future works are presented in section VI.

II. Related works

In general, multi-view human action recognition methods can be classified into two parts, conventional computer vision approach, and deep learning approach. While the former separates the process of feature extraction and features discrimination, the later puts both of them in a single pipeline.

978-1-5386-5065-3/18/$31.00 ©2018 IEEE

Chaaraoui *et.al* [1] characterized human body configuration as contour points of human silhouette and its center mass. By computing Euclidian distance between each contour point and the center of mass, they obtained the global representation of each pose. Multi-view recognition was achieved by concatenating distances signals from all views by frame-by-frame. And to predict an action they employed nearest neighbor algorithm with dynamic time wrapping (DWT) to compute distance among sequences of key poses.

Similar to [1], [5] also used silhouette in their approach. They combined silhouette with optical flow to form a local descriptor from a sequence of images and computed summary motion from 15 frames that they divided into three parts, past, current, and future to acquire motion context. Action recognition was realized by inputting the final features consisting local descriptor and motion context to 1-Nearest Neighbor with Metric learning.

Different from the previous, [2] represented sequences of images as volumetric data. The volumetric data was represented as blocks of a histogram of oriented 3D spatial-temporal gradients. The input for classier was obtained by computing distance between blocks of orientation histogram. To classify action from multiple views, they utilized multiple local classifiers to predict single action from each camera and combined the result using product rule to achieve final output.

Another work using 3D information is by Pehlivan *et.al* [3] where 3D pose was encoded as a circular model that was constructed from multiple sequences of images. The model consisted of three features (i) number of circles, (ii) the area of the outer circle, and (iii) the area of the inner circle. Classification task was performed by using distance metric algorithm with Euclidian distance.

As mentioned before, multi-view human activity recognition also can be solved using DNN approach. [7] proposed a DNN model that consisted of three convolutional layers in the first block, two dense layers in the second, two LSTM in the third block, and softmax classifier in the last layer. The input of network was a gray-scale image with a size of 256x256. Evaluated on multicamera driving simulator dataset and West Virginia University multiview action recognition dataset, the model showed high accuracy rate. However, it should be noted that in both datasets the number of subjects is quite small. Moreover, the author did not perform cross subject validation then its generalization performance is not clear.

In this paper, we propose a DNN model to estimate human activity from multiple views. The model was evaluated on challenging benchmark dataset, IXMAS dataset [4], with one subject cross-validation protocol. Detail of the model is explained in the following section.

III. Proposed Model

In this section, we describe the detail of our proposed DNN model. The model includes three main parts, CNN blocks, MSLSTMRes, and dense layer with softmax activation function [6] that has 11 output units (see Fig. 1). Spatial information is extracted in the CNN block while temporal information extraction and classification are performed by MSLSTMRes and Dense layer. In this model, late data fusion is performed, in which data from each camera is processed separately through the CNN block and MSLSTMRes and are concatenated before the dense layer. Therefore, the final layer does not perform averaging prediction score that treats each feature from each camera independently, instead, it attempts to relate information from all cameras to classify an action.

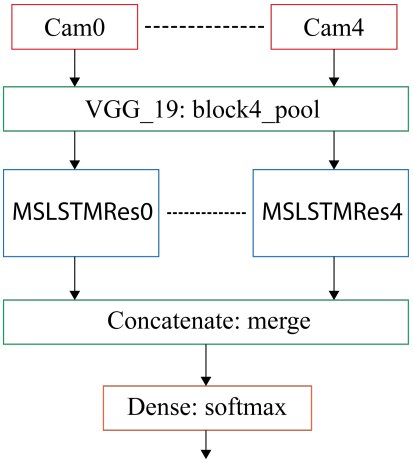

Fig. 1: Structure of our proposed model. The structure consists of three parts: CNN block, MSLSTMRes, and Dense layer. The model has five inputs units that represent images from five cameras.

A. Pre-trained VGG-19

Since to train CNN block from scratch requires large dataset, in the proposed model we utilized pre-trained VGG-19 model [8] trained on ImageNet and extract spatial features from its intermediate layer (block4_pool). The block produces a feature map f of shape $H \times W \times C$, where H is height, W is width, and C is channel. Hence, for T time-step the feature vector is

$$F = [f_1, ..., f_T], \quad \text{where } f_t \in \mathbb{R}^{H \times W \times C} \tag{1}$$

For simplicity, from this part we denote the dimension of a feature map as $\mathbb{K}$.

B. Average of soft attention model

In the proposed model, soft attention network is employed to retrieve information about the area of moving object. Given a feature vector F of shape $T \times \mathbb{K}$, the area of subject's movement can be realized by averaging attention score map over T.

The first step to calculate the average of the attention score map is to estimate attention probability at each time step. For feature map at t-th time step f_t, the attention probability is given by

$$s_t = g_{\text{att}}(f_t; \theta_t) \tag{2}$$

$$\alpha_t = \text{softmax}(s_t) \tag{3}$$

where g_{att} is attention network with weight θ_t and s_t is attention score map for the given feature map. The attention score

α_t is probabilities produced by softmax function that contains the subject of interest with the higher probability compared to the rest. The attention network can be any network, in the proposed model we utilized feed-forward neural network.

After obtaining attention probability at each time step, the next step is to estimate the average of the attention probability and to use it to compute the attended features, in this case, the area of subject movement. The average of attention probability and the attended features are given by

$$A = \frac{1}{T}\sum_{t=1}^{T}\alpha_t \tag{4}$$

$$\hat{f}_t = f_t \odot A \tag{5}$$

where $\odot$ represent element-wise operator or Hadamard product [9]. A is average attention probabilities over time, and $\hat{f}_t$ is the attended feature at t-th time step.

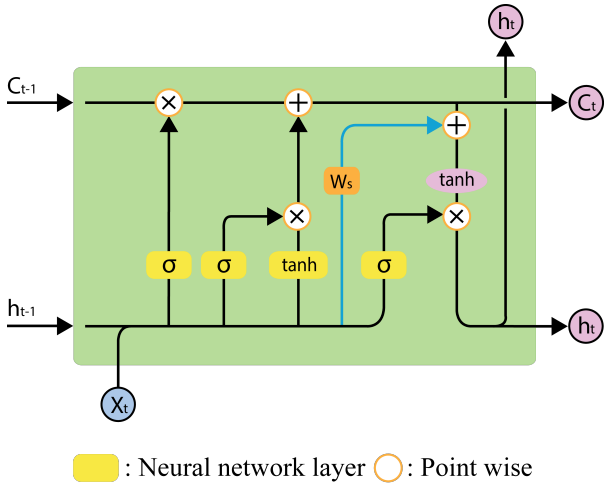

Fig. 2: Implementation of residual learning in LSTM by performing shortcut connection after forgetting old information and adding the new one.

C. Residual learning

In its original paper [10], residual learning was designed to address the degradation problem in deep neural networks, where deeper network faced saturated accuracy rate. Different from highway network [11], He *et.al*'s formulation always opens its identity shortcut, thus residual function is always learned [10]. Residual learning can be expressed as:

$$y = \mathcal{F}(x, \{W_i\}) + W_s x \tag{6}$$

where x and y represent input and output, $\mathcal{F}(x, \{W_i\})$ is the residual mapping to be learned, and W_s is a linear projection that is used when the dimension between $\mathcal{F}(x, \{W_i\})$ and x is not equal, which can be realized by employing linear mapping.

D. Revising residual learning in LSTM

Long short-term memory (LSTM) [12] was proposed to solve the problem of vanishing and exploding gradient, faced by conventional recurrent neural networks (RNN) [6]. By incorporating a memory cell to its structure, LSTM allows RNN to decide when to forget old information and when to update its content based on given new information. LSTM is able to keep important features, learned from the previous stages, as it does not overwrite its content at each time step. For given inputs x_t, h_{t-1}, and C_{t-1}, LSTM at t-th time step can be formulated as:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{7}$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{8}$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{9}$$

$$\tilde{C}_t = \sigma(W_{\tilde{C}}[h_{t-1}, x_t] + b_o) \tag{10}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{11}$$

$$h_t = o_t \odot tanh(C_t) \tag{12}$$

i_t, f_t, and o_t represent the input gate, forget gate, and output gate of LSTM. While $\tilde{C}_t$, C_t, and h_t are modulated input, memory cell, and hidden units.

Fig. 2 exhibits the structure of LSTM with residual learning. Different from [13], which adds the shortcut to hidden units, our residual LSTM performs shortcut connection when computing memory cell. Therefore, we do not need to rescale the main output as it will be transformed using tanh function. Moreover, our implementation still satisfies the basic form of the residual function $\mathcal{F}$ that requires two or more layers of network to take advantage of residual learning. The update equation for our residual LSTM can be written as follow:

$$h_t = o_t \odot tanh(C_t + W_s x_t) \tag{13}$$

E. MSLSTMRes topology

In this paper, we also attempt to investigate the effect of using ensemble topology in LSTM. Different from our baseline model (Fig. 3) that consists of two layers LSTM with 512 output units, in MSLSTMRes (Fig. 3) we replace the first LSTM with multiple stacked LSTMRes with smaller output units. Such structure reduces the total number of parameters in our network by half. This topology is designed to capture local and high abstracted temporal features of data. All hypothesize from the blocks are concatenated before being passed to the final layer.

IV. Experiments

A. Data

In this paper, we evaluated the proposed model on IXMAS dataset [4]. The dataset has been widely used for multi-view human activity discrimination by many researchers [1]–[5]. It consists of videos from 12 subjects performing 13 actions. The actions include check watch, cross arms, scratch head, sit down, get up, turn around, walk, wave, punch, kick, point, pick up, and throw. We did not use point and throw actions in our experiments since they were not performed consistently by all subjects. The videos were recorded using five cameras at 23 fps. Each subject performed every action three times and chose position and orientation freely. Each frame is an RGB image with a size of 390x291 pixels, but in this paper, it was resized to 73x73 pixels. In the preprocessing step, normalization using mean subtraction and standard deviation division

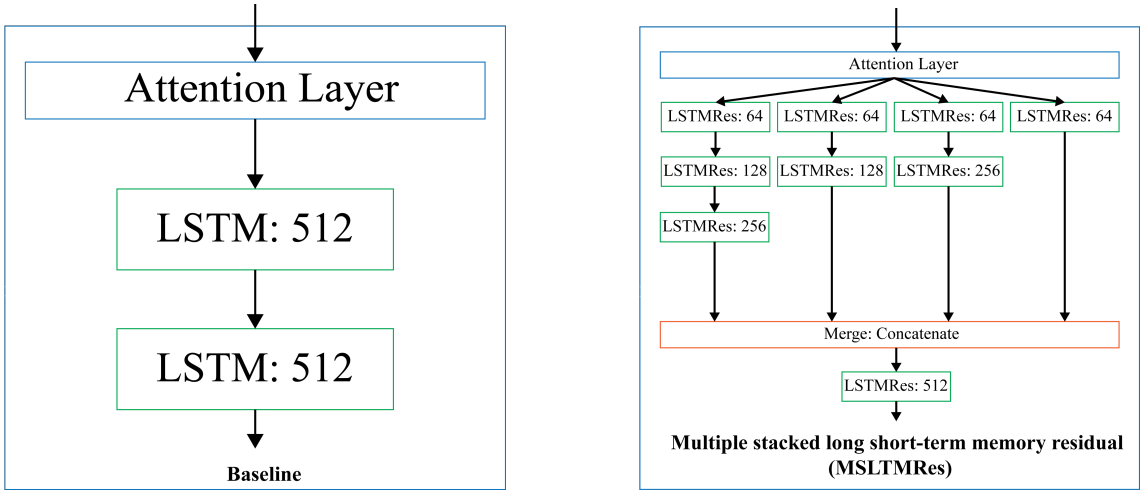

Fig. 3: The structure of baseline model (left) and proposed MSLSTMRes (right). In both models, the final LSTM produces non-sequence outputs, while the previous ones yield sequence outputs.

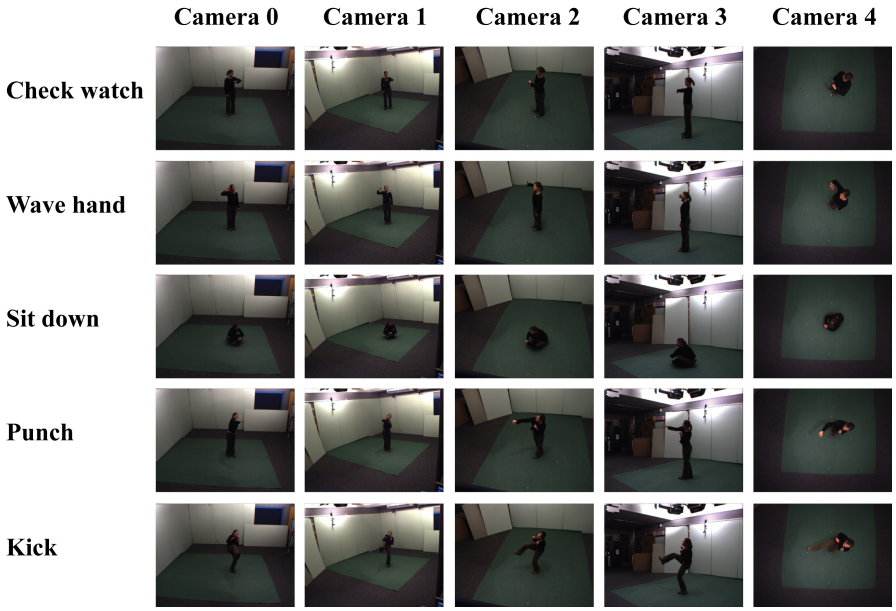

Fig. 4: Examples of IXMAS dataset [4] of a person performing check watch, wave hand, sit down, punch, and kick actions from 1st, 2nd, 3rd, 4th, and 5th cameras.

was performed at each channel of each image. Moreover, in this experiment, a sequence of images of every action was trimmed to 20 frames. Hence, for an action, the size of data is 5x20x73x73x3.

B. Methods

Proposed model was evaluated using 12-fold one-leave-subject cross-validation, in which for every fold one subject was used as test data and the remaining as training data. In the training process, the order of the data was shuffled randomly to improve the generalization performance of the model. The model was trained using RmsProp algorithm [14] with learning rate 1.01e-04. And in LSTM and LSTMRes part we set the value of dropout and recurrent dropout [15] to 0.3 and 0.5. The value of learning rate and dropout were obtained by performing random search and grid search using subsample data. For each fold, the model was trained within ≈ 100 iterations with batch size 20. Furthermore, to compute the error loss between predicted and actual output, categorical cross entropy [6] was employed.

Since this paper not only proposes a DNN model for multi-view action recognition but also investigates the effect of ensemble structure with LSTM, we also conducted an experiment using stacked LSTM (baseline) in the place of MSLSTMRes. The baseline structure was trained using the same parameter values, with the only difference being the batch size that was set to 15 for this model due to hardware limitations; the baseline model has much more parameter compared to MSLSTMRes.

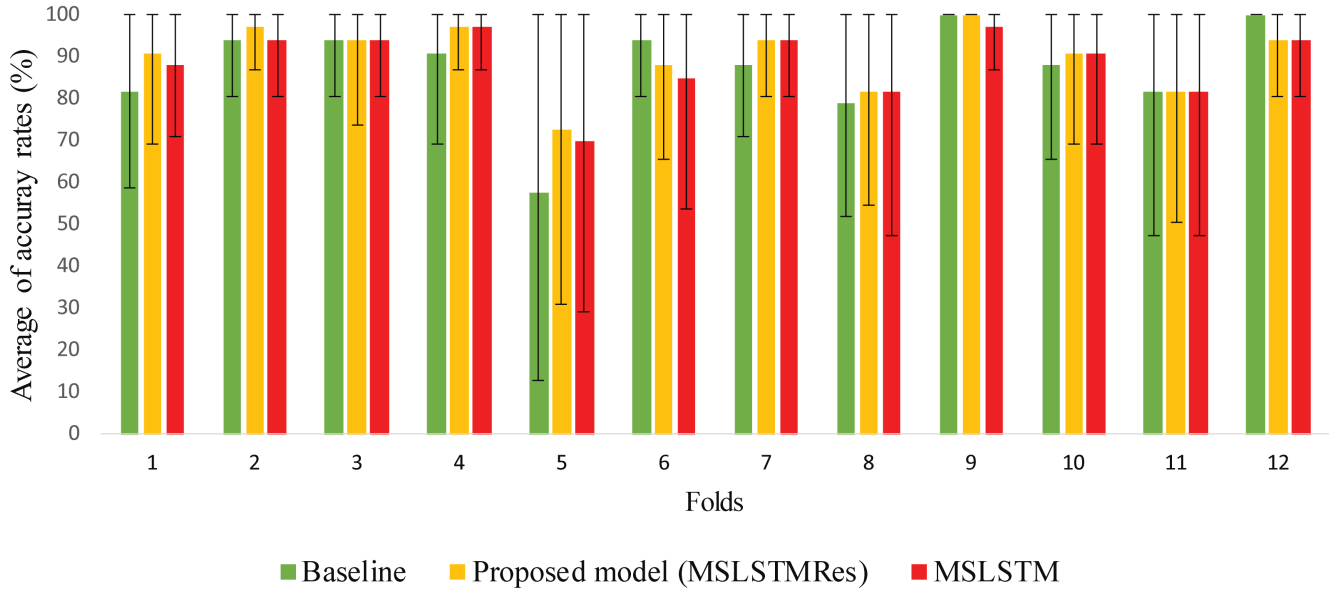

Fig. 5: The comparison result among stacked LSTM (baseline), MSLSTM, and MSLSTMRes for each fold. The accuracy rate is the average of true positive for each action.

	Check Watch	Cross Arms	Scratch Head	Sit Down	Get Up	Turn Around	Walk	Wave Hand	Punch	Kick	Pick Up
Check Watch	77.78	2.78	8.33			2.78			5.56	2.78	
Cross Arms	8.33	77.78	8.33			2.78			2.78		
Scratch Head	8.33	8.33	72.22			2.78		5.56	2.78		
Sit Down				100							
Get Up					100						
Turn Around						97.22		2.78			
Walk							100				
Wave Hand	8.33	2.78	2.78			5.56	2.78	72.22	5.56		
Punch	11.11		2.78			2.78		5.56	77.78		
Kick						5.56	2.78		5.56	86.11	
Pick Up											100

	Check Watch	Cross Arms	Scratch Head	Sit Down	Get Up	Turn Around	Walk	Wave Hand	Punch	Kick	Pick Up
Check Watch	83.33	2.78	2.78			2.78			8.33		
Cross Arms		91.67	5.56						2.78		
Scratch Head	8.33	11.11	80.56								
Sit Down				100							
Get Up					100						
Turn Around						97.22	2.78				
Walk							100				
Wave Hand	8.33	2.78	8.33			2.78	2.78	66.67	8.33		
Punch	8.33	2.78	5.56			2.78			77.78		2.78
Kick							2.78		2.78	94.44	
Pick Up											100

Fig. 6: Confusion matrices for baseline model (left) and proposed model (right) for classification task on 12 actors and 11 actions. The red blocks mark misclassification rates that are higher than 5%.

V. Results

A. Comparison of MSLSTMRes and baseline

This section explains the experimental result of the baseline and the proposed model evaluated with IXMAS dataset. The details of the results are explained in Fig. 5. From the graph, it can be seen that in most folds MSLSTMRes outperforms the baseline model, except in the 6th and 12th fold, in which the baseline achieves slightly higher recognition rate. On average, the baseline model reached an accuracy rate of $87.37 \pm 11.61\%$, while the proposed model obtained a recognition rate of $90.15 \pm 7.87\%$. However, it should be noted that in 5th fold both models showed an accuracy no more than 75%. This contrasts with the result in the 9th fold, where both models obtained 100% recognition rate.

Fig. 6 shows the confusion matrices of both models for each action. The baseline model obtained unsatisfying recognition rates for check watch, cross arm, scratch head, wave, and punch. The model mostly misclassified actions involving hand movement such as check-watch action. That can be the result of similar body configuration produced by those actions. Similar to the baseline model, the proposed model also faced difficulty to recognize actions performed with hands. However, for check watch and cross arms, the proposed model achieved better results.

One interesting point from the result is that both models often misclassified other actions performed with hands as check watch. Two possible reasons are the similar body configuration among those actions and the small resolution size of the images. Hence, it is difficult for the models to observe the difference of arm poses among the actions. Besides, it should be noticed that check watch action requires a subject to move his hand to position that lies among the other actions; i.e. it can become ”scratch head” when the person lifts up his arm, or ”punch” if he stretches his arm, or ”cross arm” if he fastens his arms.

TABLE I: Comparison with methods using similar protocol on IXMAS dataset [4].

Approach	Input	Actors	Rate
Pehlivan *et.al* [3]	Encoded volumetric data	10	90.91%
Chaaraouia *et.al* [1]	Silhouette	11	85.86%
Weinland *et.al* [2]	3D HOG	11	83.5%
Tran,and Sorokin [5]	Silhouette and Optical Flow	12	81%
Our method	Sequences of Raw Images	12	90.15%

B. MSLSTM vs MSLSTMRes

In this part, the improvement of representing residual learning in LSTM is explained. Comparison between LSTM and LSTMRes was conducted with the same experimental protocol. As shown in Fig. 5, it can be seen that introducing residual learning improved LSTM performance in 1st, 2nd, 5th, 6th, and 9th folds. Although improvement did not occur in all cases, it can be stated that LSTMRes may produce better or at least similar performance with LSTM. The average improvement achieved by LSTMRes is 1.26%.

C. Comparison with previous methods

To clarify the improvement of the proposed model we compare the model with the result of other approaches, utilizing the same dataset and similar experiment protocol. However, it should be noted that we do not reproduce their result and we utilized different kinds of input data.

Table I presents the detailed comparison between our approach and the previous methods. Despite our method utilized sequences of raw image as its input, its recognition rate outperforms other methods utilizing expert-designed features extractor and more informative features such as silhouette and optical flow. Nevertheless, compared to the method using volumetric information [3] our method has slightly lower recognition rate. That can be related to the quality of data. As explained before, in our approach we merely used sequences raw images that were resized to the size of 73x73. Furthermore, it should be pointed out that Pehlivan *et.al* [3] did not evaluate their method on 12 subjects but only on 10 of them.

VI. Conclusions

In this paper we propose a DNN model for multi-view human activity recognition. The model consists of three parts, CNN block, MSLSTMRes, and dense layer. And it was evaluated using IXMAS dataset with one-subject-cross-validation. Although the model only utilizes sequences of raw images as its input, it outperforms previous methods that utilize more informative features such as silhouette and optical flow. On average, our proposed model achieved $6.72 \pm 3.43\%$ higher accuracy rate compared to those methods. Furthermore, from the experimental result, we found that employing multiple stacked LSTMRes can improve the performance of the baseline model about 2.78%.

Despite promising result obtained by our proposed model yet there are still many possible future works that can be done to clarify the performance of the proposed model and to improve its performance. Our future works include (i) performing temporal evaluation in online scenario, (ii) evaluating the proposed model with other multi-view action datasets e.g. MuHaVi [16], NIXMAS [2], and i3DPost [17], and (iii) employing capsule networks [18] to allow the model to select relevant features from certain cameras in predicting an action.

VII. Acknowledgments

This work was supported by JSPS Kakenhi under Grant No. 26285212.

References

[1] A. A. Chaaraoui, P. Climent-Pérez, and F. Flórez-Revuelta, "Silhouette-based human action recognition using sequences of key poses," *Pattern Recognition Letters*, vol. 34, no. 15, pp. 1799–1807, 2013.

[2] D. Weinland, M. Özuysal, and P. Fua, "Making action recognition robust to occlusions and viewpoint changes," in *European Conference on Computer Vision*. Springer, 2010, pp. 635–648.

[3] S. Pehlivan and P. Duygulu, "A new pose-based representation for recognizing actions from multiple cameras," *Computer Vision and Image Understanding*, vol. 115, no. 2, pp. 140–151, 2011.

[4] D. Weinland, R. Ronfard, and E. Boyer, "Free viewpoint action recognition using motion history volumes," *Computer vision and image understanding*, vol. 104, no. 2, pp. 249–257, 2006.

[5] D. Tran and A. Sorokin, "Human activity recognition with metric learning," *Computer Vision–ECCV 2008*, pp. 548–561, 2008.

[6] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[7] R. Kavi, V. Kulathumani, F. Rohit, and V. Kecojevic, "Multiview fusion for activity recognition using deep neural networks," *Journal of Electronic Imaging*, vol. 25, no. 4, pp. 043 010–043 010, 2016.

[8] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[9] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge university press, 1990.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[11] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Highway networks," *arXiv preprint arXiv:1505.00387*, 2015.

[12] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[13] J. Kim, M. El-Khamy, and J. Lee, "Residual lstm: Design of a deep recurrent architecture for distant speech recognition," *arXiv preprint arXiv:1701.03360*, 2017.

[14] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural networks for machine learning*, vol. 4, no. 2, pp. 26–31, 2012.

[15] Y. Gal and Z. Ghahramani, "A theoretically grounded application of dropout in recurrent neural networks," in *Advances in neural information processing systems*, 2016, pp. 1019–1027.

[16] S. Singh, S. A. Velastin, and H. Ragheb, "Muhavi: A multicamera human action video dataset for the evaluation of action recognition methods," in *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*. IEEE, 2010, pp. 48–55.

[17] N. Gkalelis, H. Kim, A. Hilton, N. Nikolaidis, and I. Pitas, "The i3dpost multi-view and 3d human action/interaction database," in *Visual Media Production, 2009. CVMP'09. Conference for*. IEEE, 2009, pp. 159–168.

[18] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Advances in Neural Information Processing Systems*, 2017, pp. 3859–3869.