A Unified Deep Framework for Hand Pose Estimation and Dynamic Hand Action Recognition from First-Person RGB Videos

Viet-Duc Le[*], Van-Nam Hoang [†], Tien-Thanh Nguyen [*], Van-Hung Le [‡], Thanh-Hai Tran[*], Hai Vu[*], Thi-Lan Le[*]

[*] School of Electronics and Telecommunications,
Hanoi University of Science and Technology, Hanoi, Vietnam
Email: lan.lethi1@hust.edu.vn
[†] MICA, Hanoi University of Science and Technology, Hanoi, Vietnam.
[‡]Tan Trao University, TuyenQuang, Vietnam.

***Abstract*—Understanding hand action from the first-person video has emerged recently thanks to its wide potential applications such as hand rehabilitation, augmented reality. The majority of works mainly reply on RGB images. Compared with RGB images, hand joints have certain advantages as they are robust to illuminations and appearance variation. However, previous works for hand action recognition usually employed hand joints that are manually determined. This paper presents a unified framework for both hand pose estimation and hand action recognition from first-person RGB images. First, our framework estimates 3D hand joints from every RGB image using a combination of Resnet and a Graphical convolutional network. Then, an adaptation of a SOTA method PA-ResGCN for the human skeleton is proposed for hand action recognition from estimated hand joints. Our framework takes advantage of efficient graphical networks to model graph-like human hand structure in both phases: hand pose estimation and hand action recognition. We evaluate the proposed framework on the First Person Hand Action Benchmark (FPHAB). The experiments show that the proposed framework outperforms different SOTA methods on both hand pose estimation and hand action recognition tasks.**

I. Introduction

Hand is one of the most crucial means that humans use to interact with the world. Hence, the task of estimating human hand pose as well as understanding hand action from images (or video) play an important role in the field of computer vision. There are many applications for these tasks ranging such as smart home devices controlling [1], rehabilitation assessment in medicine [2].

In those applications, visual data can be captured using an ambient camera (mounted at a fixed position in the environment) or a wearable camera (mounted at a certain position on a human body). While ambient cameras can capture the entire scene including the human body and hand movement, they are limited to the camera field of views and difficultly scale to a larger environment. A wearable camera (mostly mounted at the forehead or the chest of a human) captures the scene n front of the human as a human eye (first-person view - FPV). By doing so, hand in action can be acquired at a fine-grained level everywhere wearer is for a better study of human hand object interaction. However, it has faced numerous challenges such as rapid changes in illuminations, significant camera motion, and complex hand-object manipulations. In general, the problems of hand pose estimation and human hand recognition are usually solved independently. Hand pose can be estimated from a single RGB image or a sequence of images while human action will be carried out directly from RGB video if the hand skeletons are unavailable. Although, some studies showed that hand pose features are the most discriminative ones for recognition among RGB, Depth, and Skeleton modalities [3]. Recently, hand pose estimation could achieve a very high precision (e.g. estimation error is less than 10mm), this motivates us to use the automatically estimated hand pose for recognition. Using skeleton information, hand action recognition methods can exploit the graph-like structure of a hand and solves it using an efficient graph convolutional network (GCN). However, the evaluation of such combined techniques is modesty investigated.

In this paper, we propose a unified framework for 3D hand pose estimation and action recognition from a first-person view. Our framework includes two modules that are easily plug-able as Fig. 1. The first module aims at estimating hand pose (3D skeleton joints) from every single RGB image. To this end, we combine Resnet for 2D hand pose prediction and a graph convolutional network for generating 3D hand joints. The second module takes resulted in 3D hand pose as input, deploys PA-ResGCN [4] model for extracting features and classifying hand action. Overall, our framework takes only monocular RGB video as the input and outputs both the sequence of the 3D hand pose and the action labels. Furthermore, this model can be applied to build practical applications using low-cost sensors (only color image data is obtained). The proposed framework will be evaluated in a First-Person Hand Action Benchmark introduced in [3] for both hand pose estimation and hand action recognition tasks.

In summary, our main contribution is two-fold: A comparative study on different combinations of CNN and GCN models for hand pose estimation and a unified framework for hand action recognition that takes automatic hand pose estimation into account and improves recognition accuracy comparing to the SOTA methods using RGB directly. In the following, we will present related works in section II, our proposed framework, and a detailed description of methods in section III. Experimental results and conclusions will be presented in sections IV and V respectively.

II. Related works

In this section, we briefly present recent works for hand pose estimation and action recognition. Hand pose estimation is the task of predicting hand (represented by a set of joints, bones,...) position in a 2D image or the simulation in a 3D space. Early works utilized the efficiency of machine learning techniques, for instance random forest and its variations [5]. The estimation task then got a breakthrough with specific advancements of deep learning and the popularity of depth cameras. Choi *et al.* [6] proposed a robust solution for accuracy 3D hand-object estimation with two networks, one for object and one for hand. Doosti *et al.* in [7] based on Graph model, presented the Hope-Net network to jointly estimate the poses of the hand and handled object both in 2D and 3D. In our work, we adopt Hope-Net as the baseline of the estimation module.

Hand action recognition is a specific case of human action recognition. The success of deep learning has led to the rise of CNN and RNN based methods for action recognition in general and hand action recognition in particular. In RNN-based methods, the skeleton of the hand is converted to sequential vectors with predefined traversal rules, then learned by RNN models such as the LSTM [8], BiRNN [9]. As for CNN's, by applying some transformation techniques, pseudo-images are generated based on the skeleton information and modeled by CNN models [10]. Recently, to precisely represent the graph-structure of the skeleton data, Yan *et al.* [11] first applied Graph Convolution Network (GCN) to the task with the ST-GCN network. As an improvement of ST-GCN, Shi *et al.* [12] proposed a novel two-stream adaptive graph convolutional network (2s-AGCN) which can adaptively learn the topology of the graph for different layers and samples. Cheng *et al.* [13] proposed a Shift-GCN network with lightweights point-wise convolutions to overcome the computational complexity of GCNs. Recently, Song *et al* [4], inspired by the success of ResNet architecture, introduced a PA-ResGCN network with a ResGCN module and a partwise attention block. Our work will take advantage of the PA-ResGCN performance to build an action recognition module.

III. Proposed method

We propose a unified framework for hand pose estimation and action recognition directly from monocular RGB video, as illustrated in Fig.1. Specifically, we focus on first-person video and actions contain hand-object interaction, which poses some challenges and has received much attention recently. Our framework consists of two parts: 3D hand pose estimation and action recognition which will be described in detail in the following sections.

A. 3D hand pose estimation from RGB Image

The objective of this module is to give an RGB image that contains a hand as an input, our network has to determine its corresponding 3D hand pose in a form $\{\mathbf{p_i}\}_{i=1}^{J}$. $\mathbf{p_i} \in R^3$ denotes the 3D locations of the i^{th} joint in a world coordinate system. This can be done directly as a regression problem or indirectly through 2D hand pose as an intermediate result, where $\mathbf{p_i} \in R^2$ represents joint in image coordinate system. In our work, we use the 21-joints hand model as shown in Fig. 2, which is popular among different datasets.

1) Baseline network: We adopt a deep graph-based model HopeNet [7] as the baseline network. It consists of three consecutively sub-network: Resnet-10 as a backbone for 2D hand pose estimation and feature extraction, GraphCNN to correct predicted 2D hand pose with graph constraints to take advantage of human hand structure, and the GraphUnet to predicts 3D hand pose from 2D hand pose. The network is trained separately at first and end-to-end with joint loss function at the end.

In this paper, we propose two network structures derived from the baseline network: (1) CNN-only networks which are traditional multi-layer CNN such as ResNet, DenseNet; (2) CNN combines with GraphCNN to perform 3D hand-pose estimation, without the use of GraphUnet as in baseline method [7].

2) CNN-only networks: In this type of network, we only use CNN for estimating 3D hand pose. The input of the network is an RGB image with the shape of 224x224x3. At the end, we replace the original FC layer, which is a regression FC layer consists of 63 neurons (21 joints x 3 dimensions) with a linear activation function.

3) CNN with GraphCNN for 3D hand pose estimation: GraphCNN is a network that consists of three layers of graph convolution, which will be explained in the next section. For this network configuration, we use CNN for initially predicting a 3D hand pose. After that, the predicted 3D hand pose will be fed into GraphCNN along with the feature from the last layer. The hand pose and feature vector are concatenated into a single vector as the input of GraphCNN. The output of the network is a corrected 3D hand pose modified from initialized hand pose from CNN. GraphCNN [7] is one type of GCN, which contains multiple graph convolution layers. The idea behind GCN is to apply convolution operator on graph data instead of an image, that operator can be defined as:

$$f(X) = \sigma(AXW) \tag{1}$$

where σ is non-linear activation function, X is matrix of input graph, W is learnable weight matrix and A is adjacency matrix. To avoid gradient vanishing or explosion problem, the authors propose to use re-normalization trick from [14], so A in Eq.1 is replace by $\widetilde{A}$:

$$\widetilde{A} = \widehat{D}^{-\frac{1}{2}}(A + I)\widehat{D}^{-\frac{1}{2}} \tag{2}$$

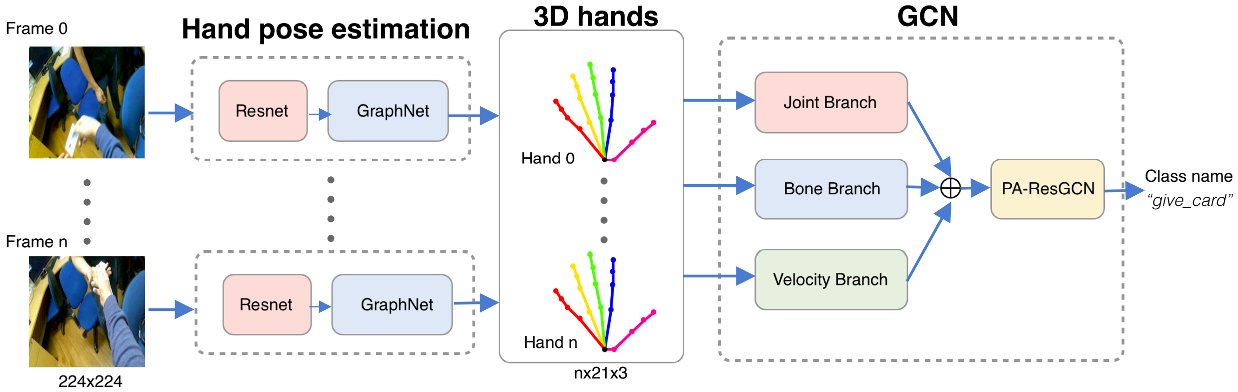


Fig. 1: Overall framework for hand pose estimation and hand action recognition.

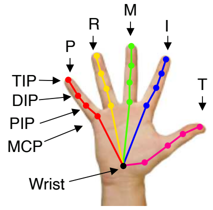


Fig. 2: The 21-joints hand model.

where $\widehat{D}$ is diagonal node degree matrix and I is identity matrix. Another differ from the original GCN is the author make A a trainable matrix, but initially create with different form: random, zeros, ones, eyes, adjacency. In our work, we also evaluate the network under different initializing methods.

B. Hand action recognition

Recently, GCN-based have shown their effectiveness for skeleton-based human action recognition. In [4], a strong neural network model based on graph convolutions named PA-ResGCN is introduced. This model leverages the use of residual for the GCN model and achieves state-of-the-art results for human action recognition. In this work, we propose to apply this model for hand action recognition.

In PA-ResGCN, instead of using only point positions, bone features and motion velocities are employed to feed to GCN. The original 3D coordinates of a hand joints sequence are a matrix of size $C \times T \times V \times M$, where C, T, V, M are the coordinates, the number of frames, the number of joints and the number of hands, respectively, is denoted as X.

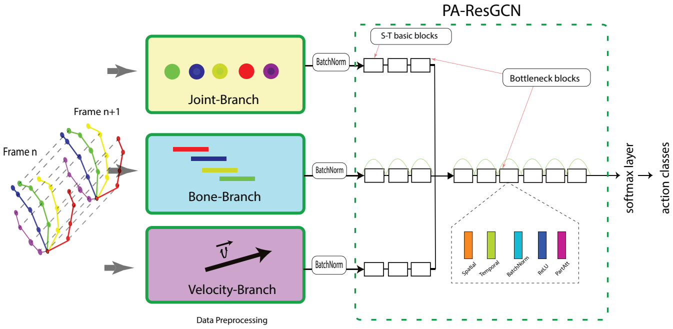


Fig. 3: Pa-ResGCN for hand action recognition.

The first part of input are joint positions that are constructed by original 3D coordinate and the relative positions. The relative joint position is $P = \{p_i | i = 1, 2, 3..., V\}$, where $p_i = x[:, :, i, :] - x[:, :, c, :]$, c is wrist joint.

The second part of input are motion velocities that are obtained by concatenating F and S, where $F = \{f_t | t = 1, 2, 3, 4, ..., T\}$, $f_t = x[:, t+2, :, :] - x[:, t, :, :]$ and $S = \{s_t | t = 1, 2, 3, 4, .., T\}$, $s_t = x[:, t+1, :, :] - x[:, t, :, :]$.

The final part of the input is bone feature that consists of the displacement of each bone $L = \{l_i | i - 1, 2, 3, 4, .., V\}$ and the bone angles $A = \{a_i | i = 1, 2, 3, 4, .., V\}$ where $l_i = x[:, :, i, :] - x[:, :, i_a, :]$, i_a is the adjacent joint of the *i-th* joint and

$$a_{i,w} = \arccos\left(\frac{l_{i,w}}{\sqrt{l_{i,x}^2, l_{i,y}^2, l_{i,z}^2}}\right) \qquad (3)$$

where w denotes the 3D coordinates.

As seen in Fig.3, after the preprocessing step, three input feature classes are fed into three batch normalization layers to normalize features. PA-ResGCN is composed of 9 convolution layers. The first three layers have three separate branches, i.e., joint branch, bone branch, and velocity branch. In the first layers, a basic block is used to process the input data, two following layers contain 2 ResGCN modules with bottleneck blocks. The input feature map of the 4th layer is acquired by concatenating three branches from previous layers. Layers 4-9th contain three ResGCN modules with bottleneck blocks in each layer and are followed by a PartAtt layer which was proposed by Yi-Fan Song et al. The purpose of PartAtt is to find the important body parts in an action sequence. In this work, we split a hand skeleton into 5 parts that are thumb, index, middle, ring, and pinky.

According to the GCN operation in [11], we adopt Eq.4 for the spatial GCN operation for each frame in a hand joint sequence.

$$f_{out} = \sum_{j=0} W_j f_{in}(\wedge_j^{-\frac{1}{2}} A_j \wedge_j^{-\frac{1}{2}} \otimes M_j) \qquad (4)$$

where the f_{in} and f_{out} denote the input and output feature maps, A_j denotes the *j*-th order adjacency matrix in the distance partitioning strategy, and normalized by $\wedge_j$. M_j and W_j are both learnable parameters. Here $\otimes$ denotes element-wise multiplication.

In the temporal dimension, as frames are consecutive, so the neighborhood of each frame is fixed at 2. A $K \times 1$ convolution is performed on the above output feature map, where K is the predefined kernel size. A basic block is created by combining spatial and temporal convolutional layers followed by the BatchNorm layer and ReLU activation layer. On the other hand, to minimize the computation time without affecting accuracy, a bottleneck block can be generated from the above basic block, which inserts two 1 × 1 convolutional layers before and after the common convolution layer to reduce the number of parameters. The ResGCN module can comprise one or more blocks mentioned above. Additionally, the dense link which is a type of residual link is also added over convolution layers and the blocks to avoid vanishing gradient.

IV. Experimental results

A. Datasets

FPHAB [3] is a first-person dataset that contains hand actions performed under different scenes and objects. It is one of the largest first-person hand action datasets with 105,459 frames, split into 45 actions and 26 types of objects under 3 indoor scenes: office, kitchen, and social. All images contain hands interacting with objects, labeled by a group of 6D magnetic sensors that can provide the accurate 3D location of each joint. This is a very challenging dataset since the number of actions is considerably large, a hand is normally occluded by an object or at the edge of the camera field-of-view. Figure 4 shows an example of FPHAB dataset. As can be seen in the picture, the hand is highly occluded with an object (a credit card) and partially visible. Moreover, there are four hands on the image but only one (bottom-right hand) is taken into account.

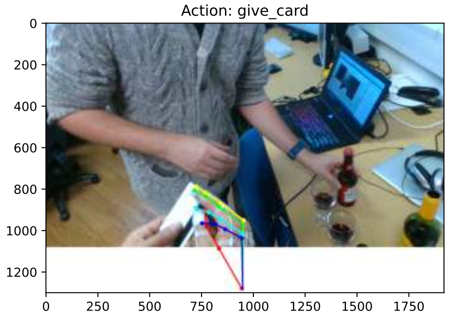


Fig. 4: An image from FPHAB dataset and its corresponding hand pose (the bottom-white bar is region outside of the image)

TABLE I: Average error of different backbone networks without using GCN

#run	Backbone	Average error (mm)
1	Resnet-10	49.98
2	Resnet-50	44.63
3	**Resnet-101**	**43.58**
4	Densenet-121	57.03

B. Experimental results

1) Hand pose estimation results: We use sequence 1 with 23,675 frames for validation, sequence 3 with 23,675 frames for testing and other sequences contain 54,680 frames to train the network. All images are resized to 224x224x3 before feeding to the network. L1 loss and Adam optimized are used for the training process. We train each network with 50 epochs and batch size = 16 frames.

To measure the accuracy of the network, we use the average L2 distance between predicted 3D hand pose and ground truth, defined as follows:

$$Err_{avg} = \frac{1}{N}\sum_{1}^{N}\frac{1}{J}\sum_{1}^{J} L2(p_i, \widetilde{p}_i) \tag{5}$$

where N and J are number of frames and number of joints (J = 21) respectively, $\widetilde{p}_i$ and p_i are predicted and ground-truth coordinates of i^{th} joint of the hand, $L2$ is the Euclidean distance between two points.

As describes in III-A, we evaluate our method with different network architectures to illustrate the contribution of each part to the final result. Table I shows the results when no graph convolution layers were added. For each network architecture, we remove the last layer and replace it with the new fully-connected contains 63 neurons (21 joints x 3 coordinate values). It can be observed in the table that Resnet is a good choice for the hand pose estimation problem and the more depth that we get, the more accurate the network presents.

After selecting a suitable backbone network, we added GraphCNN and GraphUNet and also evaluate the network with different adjacency matrix initialization methods.

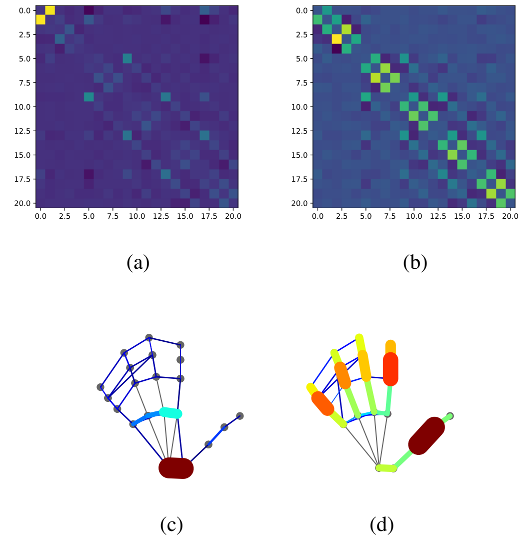
Table II demonstrates the effectiveness of each network components. With the first test using GraphUNet, we follow HopeNet [7] architecture as a baseline, which produces 2D hand first, correct it with GraphNet, and use GraphUNet to learn 2D-3D transformation. As can be seen in the table, the use of GraphUNet did not show improvement in the final result. On the other hand, the initialization method of adjacency matrix shows a strong impact on the result where identity matrix initialization gives the lowest average error with 36.6 mm. This means that network can automatically find the optimal relationships between hand joints without knowing about hand structure. This is an important feature that can be extended to other problems such as human pose estimation.

Figure 5 visualizes the learned adjacency matrix with two best initialization methods: identity matrix and skeleton (meaning assign adjacency matrix follows the hand structure at the

TABLE II: Comparison between different network combinations and adjacency matrix initialization methods

Backbone	#run	GraphCNN	GraphUNet	Init. method	Avg. error (mm)
Resnet-101	5	✓	✓	identity	41.72
	6	✓		**identity**	**36.6**
	7	✓		skeleton	39.28
	8	✓		ones	55.45

beginning of the training process). Surprisingly, the network can found a reasonable connection between joints that are useful for hand pose estimation (the bottom row in the figure). And in both cases, the thumb plays an important role to define a hand pose (the thickest edge in the figure).



(a)

(b)

(c)

(d)

Fig. 5: Learned adjacency matrices, left column: identity initialization, right column: hand structure initialization, first row: adjacency matrices after remove diagonal (self-connection or node's degree) values; second rows: visualize top 30 largest pairs, the thicker the edge is, the bigger adjacency value

Finally, we show some predicted hand poses and their corresponding input image in Fig. 6. As can be seen in the figure, our network struggle with cases where only a small part of the hand is visible or even disappears from the image field of view (first column, bottom row) in the figure. If most of the hand is in the picture (cases when hands are in the center of image), the network can predict the hand shapes with high accuracy (such as the first row in the figure).

2) Hand action recognition results: To evaluate the proposed method, we use FPHAB dataset and follow the protocol introduced in [3]. There are four protocols. The first three differ from each other by the ratio used for training and testing. The ratios are 3:1, 1:1, 1:3 for training and test respectively. For the ratio of 3:1, 3rd sequence in each action used as a test set while the remaining used for the training. Similarly, for the ratio of 1:1, the 1st and 3rd sequences are test set; while for the ratio of 1:3, the test set is the 1st, 3rd, and 4th sequences. Besides, recently, the authors of the dataset have provided the normalized version where joint coordinates are normalized by the wrist joint with a ratio of 600:575 (600 action sequences for training and 575 sequences for testing). Therefore, we also evaluate the proposed method with this protocol.

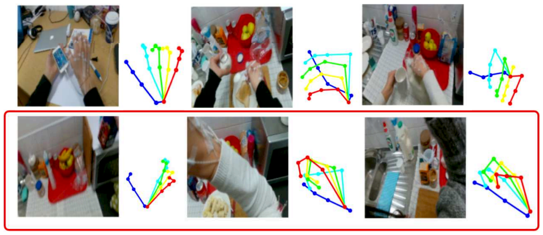


Fig. 6: Hand pose estimation results: success examples (first row) with average error $\leq$ 3mm and failure cases (second row, in red rectangle) with average error $\geq$ 300mm, we only show x,y coordinates of predicted hand poses

Two experiments are conducted. The first experiment aims at evaluating the performance of hand action recognition while the second one is to evaluate the robustness of the unified framework. Therefore, in the first experiment, the ground-truth hand joint coordinates are used while the second one employs the estimated joint coordinates by using the proposed method. In two experiments, as the number of frames in an action sequence varies, we normalize the number of frames of all samples to 100 by filling the value of missing frames by the last frame. The temporal window size K is set to 9 while the strides of the 4th and the 7th temporal convolution layers are 2. We trained the network using SGD optimizer in 50 epochs with a batch size of 30, a learning rate of 0.1, and decays by 10 at the 20th epoch.

The results of the first experiment are shown in Tab. III. Compared with the method in [3], the proposed method outperforms this method on all protocols. The accuracy obtained by the proposed methods for the first three protocols are 76.57%, 2.59%, and 90.37% while those of [3] are 58.75%, 78.73% and 84.82% respectively. It is quite interesting to see that the proposed method can learn to represent the hand action better than the method based on LSTM when few training samples are available in the 1:3 protocol. For this protocol, the proposed method obtained 17.82% of improvement over the method based on LSTM. When more samples are used, both methods can improve the recognition accuracy.

In the second experiment, to evaluate the performance of the unified framework, we employ joints coordinates that are automatically estimated by the hand pose estimation method presented in Section III.B. We also compared the recognition accuracy when applying the state-of-the-art method that is HopeNet for hand joint estimation. The obtained results are shown in Tab. IV. we can observe that thanks to the high-quality joint estimation of the proposed method, action recognition accuracies of the proposed unified framework are higher than those obtained when using the HopeNet method.

With the best configuration, the framework obtained 72.7%, 74.77%, and 79.63% accuracies for 1:3, 1:1, and 3:2 protocols respectively. Of course that these results are lower than those obtained with ground-truth joints. However, the margin is relatively small (2.87% for 1:3 protocol). This shows the potential of the application of using hand pose from RGB image for action recognition.

TABLE III: Recognition accuracy (%) on FPHAB dataset when using ground truth hand joints with different protocols

Method		Protocol		
	1:3	1:1	3:1	Norm. Data
LSTM 3D-GT [3]	58.75	78.73	84.82	_
The proposed method	**76.57**	**82.59**	**90.37**	**82.78**

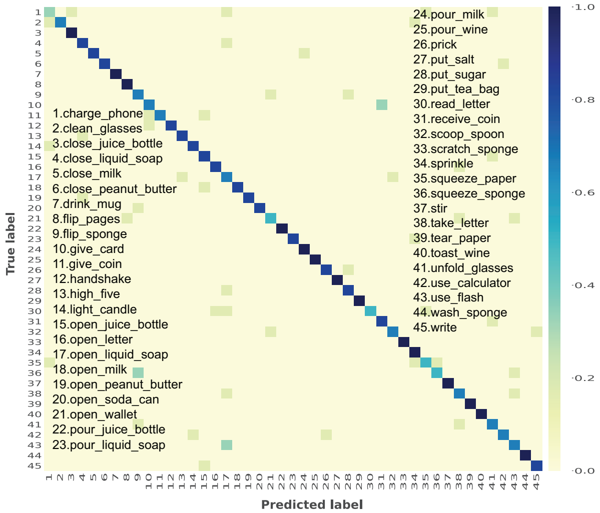


Fig. 7: The confusion matrix obtained by the proposed method for 3:1 protocol.

Figure 7 shows the confusion matrix obtained by the proposed method when using Resnet-101 and GraphCNN for skeleton initialization and estimation and 3:1 protocol. It is worth noting that the proposed method is able to recognize hand activities with large movement achieve *e.g.*, write, sprinkle, prick, pour milk. On the other hand, the performance obtained for activities with hand poses that are not changing too much is still poor.

TABLE IV: The accuracy of the proposed method for FPHAB dataset with estimated hand joints

Methods	**Protocols**		
	13	1:1	3:1
Run #5 (HopeNet [7])	67.76	70.87	77.04
Proposed method			
Run #3 (cf. Tab. II)	70.16	74.03	78.89
Run #7 (cf. Tab. II)	69.41	**75.32**	**80.74**
Run #6 (cf. Tab. II)	**73.70**	74.77	79.63

V. Conclusions Future works

In this paper, we have proposed a general framework that utilized Resnet, GraphCNN, and Res-GCN for hand pose estimation and activity recognition problems. The proposed framework outperforms the baseline methods by a large margin and achieves comparable performance when using predicted pose for activity recognition, make our system a unified framework for hand-object activity recognition through hand pose as an intermediate result. We have shown that the network can automatically exploit the hidden structure of the hand by using GCN so that our framework can be applied for other applications such as human pose estimation. In the future, we plan to deeper investigate the hand-object relationship and dealing with heavy occlusion by using temporal smoothness as well as test our framework with other datasets.

VI. Acknowlegements

This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.01-2019.315.

References

[1] T. Wu, Y. Yuan, H.-S. Yeo, A. Quigley, H. Koike, and K. M. Kitani, "Back-Hand-Pose: 3D Hand Pose Estimation for a Wrist-worn Camera via Dorsum Deformation Network, year = 2020," in *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*.

[2] C. M. Light, P. H. Chappell, and P. J. Kyberd, "Establishing a standardized clinical assessment tool of pathologic and prosthetic hand function: Normative data, reliability, and validity," *Archives of Physical Medicine and Rehabilitation*, vol. 83, no. 6, pp. 776–783, 2002.

[3] G. Garcia-Hernando, S. Yuan, S. Baek, and T.-K. Kim, "First-person hand action benchmark with rgb-d videos and 3d hand pose annotations," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 409–419.

[4] Y.-F. Song, Z. Zhang, C. Shan, and L. Wang, "Stronger, faster and more explainable: A graph convolutional baseline for skeleton-based action recognition," in *Proceedings of the 28th ACM International Conference on Multimedia*, ser. MM '20, 2020, p. 1625–1633.

[5] C. Xu and L. Cheng, "Efficient hand pose estimation from a single depth image," in *European Conference on Computer Vision*, pp. 3457–3463.

[6] C.-N. C. Chiho Choi, Sang Ho Yoon and K. Ramani, "Robust hand pose estimation during the interaction with an unknown object," in *ICCV*, 2017, pp. 3123–3132.

[7] B. Doosti, S. Naha, M. Mirbagheri, and D. J. Crandall, "Hope-net: A graph-based model for hand-object pose estimation," in *CVPR 2020*, 2020, pp. 6607–6616.

[8] D. X. Jun Liu, Amir Shahroudy and G. Wang, "Spatio-temporal lstm with trust gates for 3d human action recognition," in *Computer Vision ECCV*, pp. 816–833.

[9] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in *CVPR*, 2015, pp. 1110–1118.

[10] Y. L.-S. X. J. Liu, Y. Wang and C. Pan, "3d posturenet: A unified framework for skeleton-based posture recognition," in *Pattern Recognition Letters*, 2020.

[11] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *AAAI*, 2018.

[12] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," *CVPR 2019*, pp. 12 018–12 027, 2019.

[13] K. Cheng, Y. Zhang, X. He, W. Chen, J. Cheng, and H. Lu, "Skeleton-based action recognition with shift graph convolutional network," in *CVPR 2020*, 2020, pp. 180–189.

[14] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016.