Multi-Labeled Human Action Recognition

1st Yongjae Lee
Department of Mechanical Engineering, Yonsei University
Seoul, Republic of Korea
yongjae.lee@yonsei.ac.kr

***Abstract*—Recognizing the in-situ context of a factory is an important issue in the manufacturing process. In recent many studies to recognize the situation with various sensors are conducted, but those are containing several problems still. In this study, we design enhanced human action estimator architecture that produces a series of short descriptive sentences from a sequence of volumetric data.**

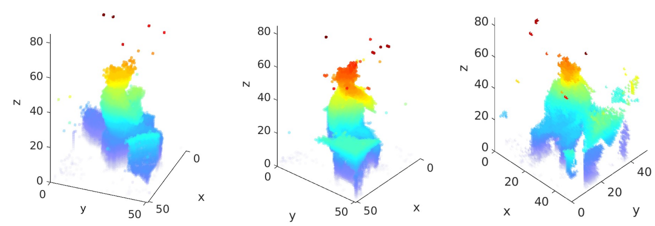
***Index Terms*—3D CNN, Human Action Recognition, Action Description**

I. Introduction

Recognizing human action by machine is important task and can be utilized in many fields. For example, in the senior care center, the safety system will alert to nurses or the patient when a dangerous action of the patient is detected by the system. In other case like a CCTV installed at a street, the public security system will protect from any latent crimes in the street. Particularly, recognizing human action technology is also utilized at manufacturing system.

Process Planning is designing a manufacturing process to make a desired product. There can be many choices to build a plan, but finding the effective plan among the candidates is the important issue. Design parameters, for example, the force or temperature or time or angle of a machining head, are the considerations to build a process plan and affect to the quality of product directly. Many researchers in this domain have involved to develop a effective planning methods. But, as we know, all of them take account into only environmental variables, not human actions. Because labor must be included at every manufacturing process, the planning methods such that do not consider the human action in manufacturing process do not hold water. Our purpose is that taking account into human action in building the process plan.

Recently, Artificial Intelligence has evolved rapidly and been adopted at many domains. For the recognizing the human action, there are many methods suggested to accomplish it. [1]–[19] Restricted in vision recognition, there are two mainstream ways: Skeleton-based and Depth-based. In Skeleton-based approach, the image of human is converted to a graph that represents bones for edges and joints for vertices. This geometrical shape and relation of each bones and joints are used as the cue to distinguish human action. This approach has advantages of invariant to scene, to human attribute, and to viewpoint. Also this approach has disadvantages: regarding with Human-object interaction and Self-occlusion issues, so that this method shows low accuracy and poor robustness. On the other hand, other studies used Depth-based approach.



(a) clap (b) drink (c) sit

Fig. 1: Three reconstructed holistic sample scene by several depth camera. The subject in sub-figure (a), (b), (c) is acting 'clap', 'drink', and 'sit' respectively.

Depth-based approach is much free from aforementioned issues, but many studies concentrated on multi-class classification, so that it is hard to distinguish comprehensive actions of human.

In this project, We study and propose a machine learning architecture of human action recognition. The goal is to make a quantified data about the human activities via Depth-based approach. First we gathered volumetric data that contain comprehensive human action. Then we build a neural network model recognizing a comprehensive action as composition of basic action components. Finally, the model is trained and tested for verifying whether it has sufficient capabilities. These quantified data enrich process planning parameters and lead to a more accurate and effective result. In the following sections, we will cover about the dataset, the model, and the experiments.

II. Research Approach

A. Dataset

We acquired 3D voxelized data of human action from Action4D [18] which is one of multi-class classification approach to recognize human action. They captured the whole scene of person by using multiple depth cameras. Then they reconstruct a holistic scene by fusing multiple calibrated depth images. Fig. 1 shows sample images.

A depth image is a set point data as known as point cloud. Basically a depth image contains a lot of point and that is too big to analyze with machine learning. To reduce the heaviness of data, the depth images are sampled as three-dimensional unit, a voxel.

To adopt this gathered data for training and testing our neural net model, we carried out additional work. Because the

Index	Class
1	clap
2	drink
3	sit
4	bend
5	stand
...	...

Index	stand	sit	left_h_hold	right_h_hold	drink	walk	bend	clap	phone_call	point
1	0	1	1	1	0	0	0	1	0	0
2	0	1	1	0	1	0	0	0	0	0
3	0	1	0	0	0	0	0	0	0	0
4	1	0	0	0	0	0	1	0	0	0
5	1	0	0	0	0	0	0	0	0	0
...	...									

Fig. 2: Each frame is re-described by multiple classes instead of single class. Each item is expressed as a vector of the classes represented by 0 or 1. Zero means the item does not belong to the class, and One means the opposite. The first three items match with Fig. 1a, Fig. 1b, Fig. 1c respectively.

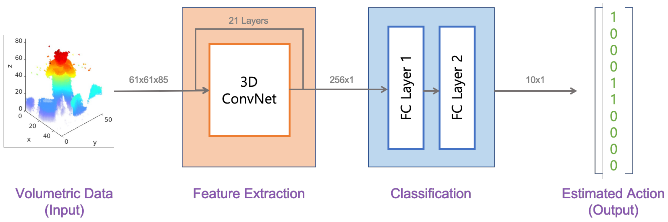


Fig. 3: An overview of classification model

given data have labels suitable for multi-class classification problems, we re-labeled it to have multiple labels suitable for multi-labeled classification problem. For example, the Fig. 1a is originally labeled as 'clap'. We rewrite the label with basic action components 'sit', 'left_h_hold', 'right_h_hold', and 'clap', since the subject is sitting while clapping which is an action of holding hands each other in a moment. The number of basic action components is ten(such that are stand, sit, left_h_hold, right_h_hold, drink, walk, bend, clap, phone_call, point). basic action components are written to each frame by one-hot encoding method. Fig. 2 shows some of conversion examples from a class to multiple classes.

B. Model

To classify a 3D voxelized data into a set of basic action components, we build a classification model. Fig. 3 shows an overview of classification model. The model consist of two main step: Feature extraction and Classification. Feature extraction is sequence of convolution layer, batch normalization layer, activation layer, pooling layer. The number of layer of feature extraction step is 21. The convolution layer multiplies the input with weights.

$$(h * f)[x, y, z] = \sum_{i,j,k} h[x-i, y-j, z-k] f[i, j, k] \tag{1}$$

where, f is input function, and h is weight function. Then, the batch normalization layer bounds the input value in the range between 0 and 1. This makes all inputs to have same expectation(Fig. 4). At the activation layer, we used ReLU(Rectified Linear Unit) function which outputs zero until $x < 0$, and increase proportional to x over 0(Fig. 5.

$$f(x) = max(0, x) \tag{2}$$

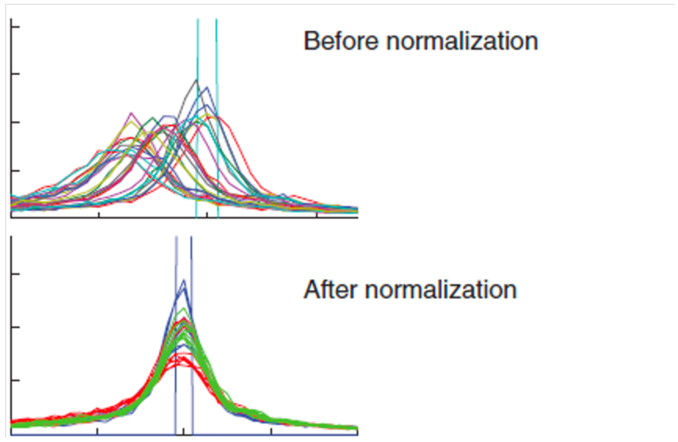


Fig. 4: The normalization works for each distribution of inputs to be the same expectation.

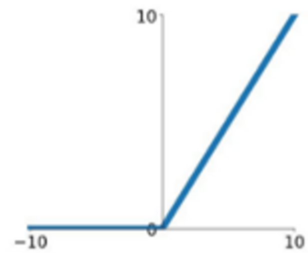


Fig. 5: Rectified Linear Unit function. This increase linearly when x over 0, but still remains at 0 when x under 0.

where, x is input and $f(x)$ is output. Pooling layer makes the spatial complexity of features to be reduced, and consequently relaxes the computation work of network. we used 2x2x2 max pooling which pick the greatest value among 2x2x2 3rd-Order tensor. Fig. 6 shows the order of layers in feature extraction step.

Classification step has two Fully connected layer which calculate probability for each basic action component. Because the number of basic action component is ten, the output of classification step is vector of size 10.

III. Result

We've trained the network 500 times and check the result. We can find the convergence tendency for loss and accuracy(Fig. 7). Loss is calculated by the distance of ground truth vector which is labeled according to II-A section and estimated value of the network. The distance measured by MSE(Mean Squared Error).

$$MSELoss = \frac{1}{N} \sum_{i}^{N} (f_i - y_i)^2 \tag{3}$$

where, f_i is estimated value of the network and y_i is ground truth value of ith data. Accuracy is measured by summing hit count for 1. The threshold of estimated value is 0.05.

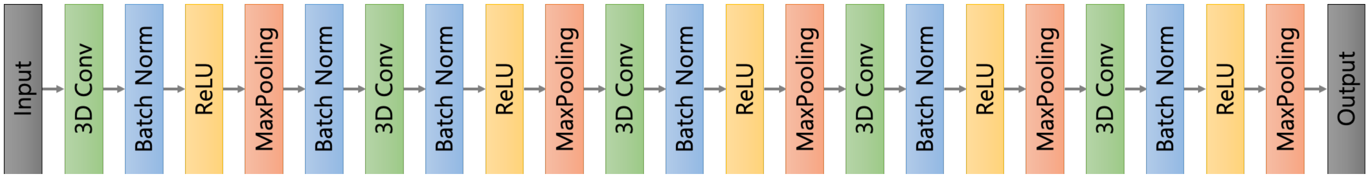


Fig. 6: The layer order of the feature extraction step. The feature extraction step consist of convolution layer, batch normalization layer, activation layer, and pooling layer.

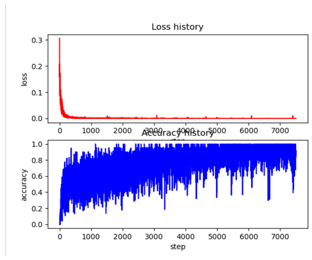


Fig. 7: Loss and Accuracy graph. According to training goes on, loss converges to 0 and accuracy converges to 1.

$$Acc = \frac{\sum_{j=1}^{M} f_{ij} + y_{ij} > 2 - threshold}{\sum_{j=1}^{M} y_{ij}} \tag{4}$$

where, f_ij is jth action component probability of ith data and y_ij is jth action component ground truth value of ith data. M is number of basic components, in our case, value is 10.

ACKNOWLEDGMENT

This study is conducted in "ESTIMATION THEORY AND APPLICATIONS" class in the 2020 first semester.

REFERENCES

[1] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN Architecture for Weakly Supervised Place Recognition," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, 2018, pp. 1437–1451.

[2] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese, *3D-R2N2: A Unified Approach for Singleand Multi-view 3D Object Reconstruction*, ser. Lecture Notes in Computer Science, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, vol. 9912. [Online]. Available: https://link.springer.com/content/pdf/10.1007%2F978-3-319-46484-8_23.pdf http://link.springer.com/10.1007/978-3-319-46484-8

[3] R. Zhang and B. Ni, "Learning Behavior Recognition and Analysis by Using 3D Convolutional Neural Networks," in *2019 5th International Conference on Engineering, Applied Sciences and Technology (ICEAST)*, jul 2019, pp. 1–4.

[4] Z. Qiu, T. Yao, and T. Mei, "Learning Spatio-Temporal Representation with Pseudo-3D Residual Networks," in *2017 IEEE International Conference on Computer Vision (ICCV)*, vol. 2017-Octob. IEEE, oct 2017, pp. 5534–5542. [Online]. Available: http://ieeexplore.ieee.org/document/8237852/

[5] H. Xu, A. Das, and K. Saenko, "R-c3d: Region convolutional 3d network for temporal activity detection," in *Proceedings of the IEEE international conference on computer vision*, mar 2017, pp. 5783–5792. [Online]. Available: http://arxiv.org/abs/1703.07814

[6] K. Hara, H. Kataoka, and Y. Satoh, "Learning Spatio-Temporal Features with 3D Residual Networks for Action Recognition," in *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, vol. 2018-Janua. IEEE, oct 2017, pp. 3154–3160. [Online]. Available: http://ieeexplore.ieee.org/document/8265584/

[7] M. Zolfaghari, G. L. Oliveira, N. Sedaghat, and T. Brox, "Chained multi-stream networks exploiting pose, motion, and appearance for action classification and detection," in *Proceedings of the IEEE International Conference on Computer Vision*, Zolfaghari2017, 2017, pp. 2904–2913.

[8] G. Singh, S. Saha, M. Sapienza, P. H. S. Torr, and F. Cuzzolin, "Online real-time multiple spatiotemporal action localisation and prediction," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3637–3646.

[9] A. Diba, V. Sharma, and L. Van Gool, "Deep temporal linear encoding networks," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 2329–2338.

[10] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt, "Sequential Deep Learning for Human Action Recognition," in *Human Behavior Understanding*, B. Salah Albert Ali and Lepri, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 29–39.

[11] C. Choy, J. Park, and V. Koltun, "Fully Convolutional Geometric Features," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 8958–8966.

[12] M. A. Uy and G. H. Lee, "PointNetVLAD: Deep Point Cloud Based Retrieval for Large-Scale Place Recognition," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, jun 2018, pp. 4470–4479. [Online]. Available: https://github.com/mikacuy/pointnetvlad.git https://ieeexplore.ieee.org/document/8578568/

[13] G. Riegler, A. O. Ulusoy, and A. Geiger, "OctNet: Learning Deep 3D Representations at High Resolutions," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jul 2017, pp. 6620–6629. [Online]. Available: http://ieeexplore.ieee.org/document/8100184/

[14] T. S. Kim and A. Reiter, "Interpretable 3D Human Action Analysis with Temporal Convolutional Networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, jul 2017, pp. 1623–1631. [Online]. Available: http://ieeexplore.ieee.org/document/8014941/

[15] J. Liu, G. Wang, P. Hu, L.-Y. Duan, and A. C. Kot, "Global Context-Aware Attention LSTM Networks for 3D Action Recognition," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2017-Janua. IEEE, jul 2017, pp. 3671–3680. [Online]. Available: http://ieeexplore.ieee.org/document/8099874/

[16] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "Ntu rgb+ d: A large scale dataset for 3d human activity analysis," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1010–1019.

[17] A. Shahroudy, T. Ng, Q. Yang, and G. Wang, "Multimodal Multipart Learning for Action Recognition in Depth Videos," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 10, pp. 2123–2129, oct 2016. [Online]. Available: https://ieeexplore.ieee.org/document/7346486/

[18] Q. You and H. Jiang, "Action4D: Online Action Recognition in the Crowd and Clutter," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 857–11 866.

[19] F. Poux and R. Billen, "Voxel-based 3D Point Cloud Semantic Segmentation: Unsupervised geometric and relationship featuring vs deep learning methods," *ISPRS International Journal of Geo-Information*, vol. 8, no. 5, 2019.