ICIC Express Letters
Part B: Applications
Volume **11**, Number **7**, July **2020**

ICIC International ©2020 ISSN 2185-2766
pp. **649–653**

ESTIMATING DESK WORK STATUS FROM VIDEO STREAM USING A DEEP NEURAL NETWORK

Megumi Kawata and Hajime Murao

Graduate School on Intercultural Studies
Kobe University
1-1, Rokkodai-cho, Nada-ku, Kobe 657-8501, Japan
Megumi.kawata@mulabo.org; murao@i.cla.kobe-u.ac.jp

Received January 2020; accepted April 2020

Abstract. *The goal of this research is to maintain an atmosphere conducive to better concentration on research and study in our laboratory. As a first step, we have constructed a system that recognizes a student's motions while sitting at a desk from movie data we filmed in our laboratory and puts predefined tag on the movements. Several Deep Learning Neural Networks (DLNNs) which can be used for this task have already been proposed, such as Recurrent Neural Networks (RNNs) like LSTM and 3-Dimensional Convolutional Neural Networks (3D-CNNs). However, we have an insufficient amount of movie data to train them. In this paper, we propose a DLNN consisting of a small number of CNNs unlike number of parameters used by RNNs and 3D-CNNs. In our DLNN a single frame extracted from movie data is treated as input to the CNN. Since students at the desk move only a little, it is thought to be possible to estimate students behavior from a small number of frames. We prepared 4 tags: "using computer", "using a smartphone", "talking with a friend" and "snoozing". We asked 4 subjects to perform these 4 actions in front of a camera. We filmed their motions in 3 fps data and we created two different datasets. In one, we extracted the person's poses from the movie, and in the other, after the extraction, we estimated the poses, converted them to structured elements and set them as input data. The tags that correspond to this data were set as training data. As a result, it was possible to obtain an accuracy of 97.5% when the number of CNNs was 5 and a human extraction dataset was used.*

Keywords: Behavior analysis, Human action recognition, 3D image sensor, Kinect, Principal component analysis

1. **Introduction.** We propose a neural network that can learn human behavior from movie data. Recently, communication between human and machine has been rapidly changing from passive reception to be interactive. As a result, our daily lives have become much more convenient. There has been much research on how to improve the communication ability of machines. However, smooth communication between humans and machines is hard to realize. There are decisive differences between human-to-human flexible communication and human-to-machine interactions. The main difference is thought to be the cognitive ability to understand the emotions of the communication partner and the surrounding environment. It is still difficult at the present time to add to the machine the state of the partner, especially the ability to infer feeling. The ability to estimate the emotion of a partner is an essential ability for smooth communication, and this research tries to approach this problem. In general, by using data such as images, movie data, voice, and heart rate, for machine learning, machines should be able to recognize human emotions and behaviors. The most frequently used technique is deep learning. It shows high performance in the field of image recognition and voice recognition. CNNs (Convolution Neural Networks) [1] are mainly used for image recognition. And in the field

DOI: 10.24507/icicelb.11.07.649

of voice recognition, RNNs (Recurrent Neural Networks) [2] that can process time series data such as sound waves and acoustic signals have attracted attention. RNNs have also achieved high performance in the field of natural language processing. Recently, 3D-CNN (3D Convolution Neural Network) [3,4] has been proposed in deep learning method applied to movie data. It generates models can expand 3-dimensional data of "width, height and time" by extending the convolution layer. This study suggests an appropriate neural network to recognize human behavior included in movie data.

2. **Related Works.** There are a lot of human recognition methods reported such as human detection [5,6], segmentation [7] and pose estimation [8-10]. The taking human data obtain from cameras and sensors, most of human recognition. And those data are used to train deep learning for understanding human behavior. Usually, those data are used such as time series data for deep learning. There is a major technique 3D-CNN. This is able to extract changing feature value according to time. Baccouche et al. [4] combined 3D-CNN with RNN to improve accuracy of human behavior recognition. They proposed a deep learning model which can learn the classification of human actions without any prior knowledge. Ji et al. [3] proposed their 3D-CNN model for human action. Their model extracts feature from both the spatial and the temporal dimensions by using 3D convolutions. Their 3D-CNN model has been evaluated with TRECVID and the KTH data sets. Results show that the 3D-CNN model outperforms compared methods on the TRECVID data, while it achieves competitive performance on the KTH data, demonstrating its superior performance in real-world environments. Recently, many researchers have made attempt to learn 3D-CNNs for recognition of human action in movie data. However, partly due to the high complexity of training 3D convolution kernels, it needs large quantities of training videos [11]. Accordingly, we propose a 2D-CNN model that is able to learn to use small quantities of human action data for human behavior recognition. When machines recognize behavior of human who is in the room, we verify whether our model does not need to focus on time base.

3. **System Architecture.**

3.1. **Architecture.** The architecture of our neural network is depicted in Figure 1. We propose a neural network composed of multiple input networks. It consists of several convolutional networks and one fully connected feedforward network. Each of CNN has 4

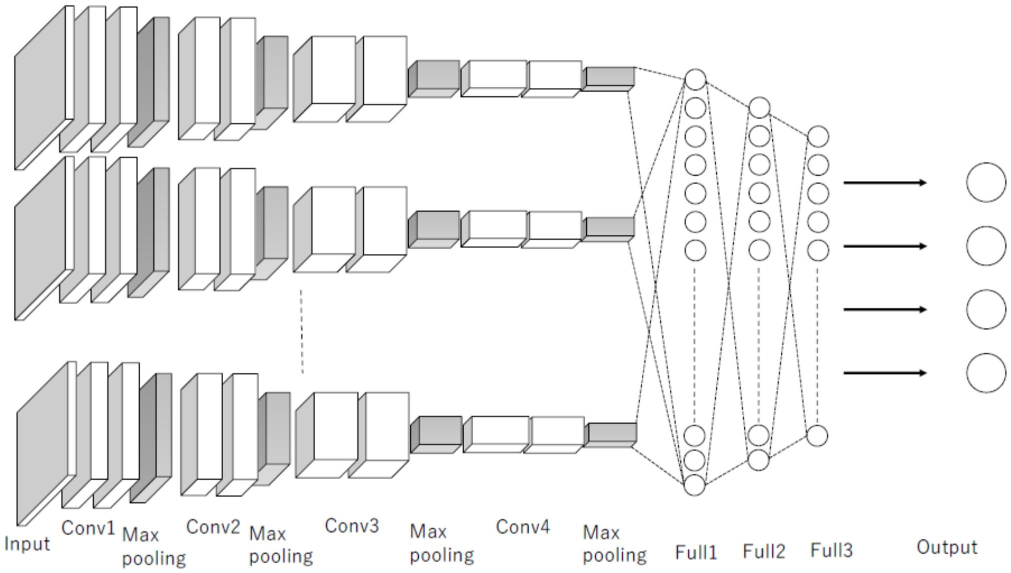


Figure 1. Our neural newtork

convolutional layers with 4 max pooling layers. We can choose the number of frames for estimation by choosing the number of CNNs. The number of CNNs is denoted by "N". For example, for N3, we use 3 CNNs. Input size is 128 × 384 × 3. After CNN, each size is 96 × 5 × 5. When CNNs were connected by using a feedforward net, the size became 96 × 15 × 5. We use ReLU as an output function for each convolutional layer and also for the output levels.

3.2. **Input layer.** In the input layer, the input data is resized to 128 px × 128 px. We use RGB data as training data and there are 3 channels. For example, for N5, we use 5 image data and 5 CNNs, and the input size is 128 × 640 × 3.

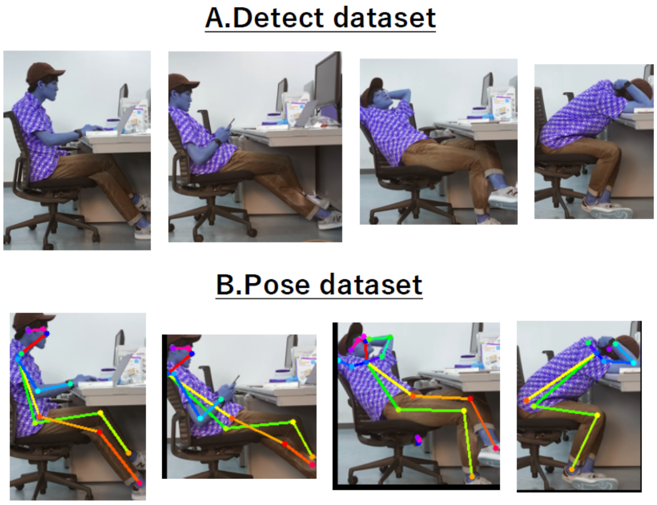
3.3. **Convolution layer.** In the convolution layer, a small feature detector called a convolution filter is used. This convolution filter scans over the image and calculates the weighted sum for each part of the image. The data read is sent to 4 convolution layers: Conv 1, Conv 2, Conv 3, Conv 4. The filter size of the convolution layer is set to 3 × 3, and the number of maps is set to Conv 1 = 128, Conv 2 = 64, Conv 3 = 32, Conv 4 = 16.

3.4. **Pooling layer.** The pooling layer is used to lower the resolution of the feature map. Through this process, the size of the map becomes smaller. This layer minimizes loss of information and reduces the amount of data. A certain degree of invariance can also be realized for positional deviation. Other features such as suppressing excessive leaning are also obtained. In this research, max pooling layers which take the maximum value of 2 × 2 combined are adopted and set.

3.5. **Fully connected layer.** In the fully connected layer, classification is performed based on the total number of features in order to identify the image. Image data, obtained by extracting a characteristic portion through a convolution layer and a pooling layer, is coupled to one node and a feature variable converted by an activation function is the output. As the number of nodes increases, the number of divisions of the feature space increases, and the number of feature variables characterizing each region also increases. Our neural network proposal constructed in our research has 3 fully connected layers of full 1, full 2 and full 3. The number of neurons in full 1 layer is 12,000, in full 2 is 4,096 and in full 3 is 1,024.

3.6. **Output layer.** We use a softmax function in the output layer. This function is effective when solving classification problems with neural network. Our neural network is classified into four classes from 0 to 3. Using a softmax function, the feature variables obtained from fully connected layers are converted into probabilities. These probabilities are then correctly classified into each region, and the classification is performed by maximizing.

4. **Experiment.** We prepared training data and test data. First, we set a camera in the lab and we shot a video of students sitting, studying in front of a desk. Then, we collected 30 fps movie data and extracted one image data in every 10 frames by using OpenCV. After that, we compiled 2 datasets from the data, A. Detect dataset and B. Pose dataset. We divided these datasets into training data and test data. Lastly, we inputted training data on our neural network. We took action data of four subjects with an RGB camera. The four subjects each acted four states for about one minute, as shown in Figure 2. We then extracted one image data from every 10 frames by using OpenCV. We made two datasets using this data. One was Detect dataset, which identified humans from the images using YOLO to extract the human figure from the image data. The other was Pose dataset. This visualized human poses from the extracted data and we used realtime multi-person pose estimation to visualize a human pose [8].



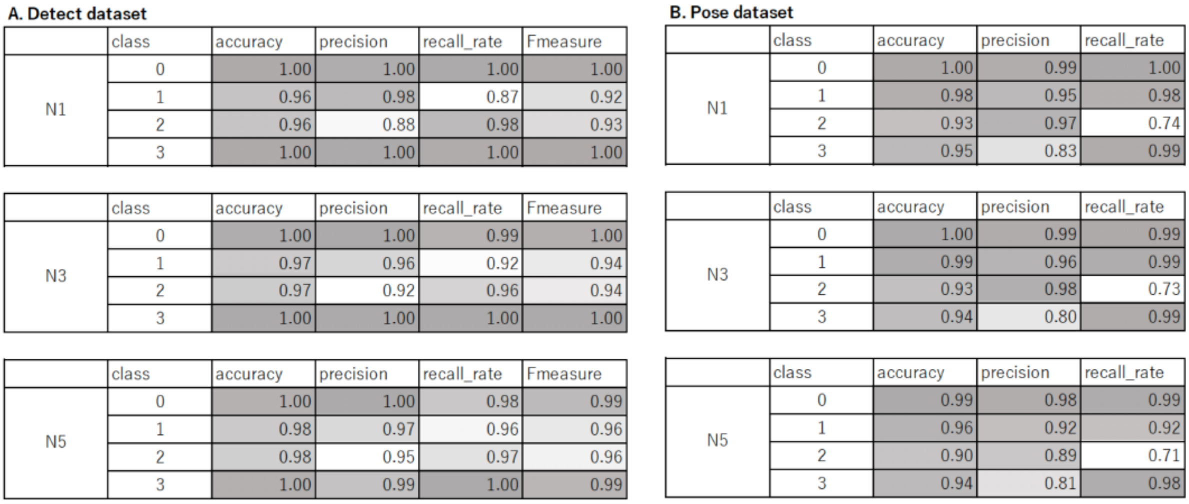
A.Detect dataset

B.Pose dataset

FIGURE 2. Two datasets

5. Results.

5.1. **Evaluation (3 subjects).** We tested the trained network with untrained data for 3 trained subjects. In Figure 3 we show the evaluations, accuracy, precision, recall rate, and F measure, which are averaged over four classes. The darker gray cells show higher values. The highest results were founded with N5 A. Detect dataset. However, most networks had fairly good results, over 90%.



A. Detect dataset

	class	accuracy	precision	recall_rate	Fmeasure
N1	0	1.00	1.00	1.00	1.00
	1	0.96	0.98	0.87	0.92
	2	0.96	0.88	0.98	0.93
	3	1.00	1.00	1.00	1.00

	class	accuracy	precision	recall_rate	Fmeasure
N3	0	1.00	1.00	0.99	1.00
	1	0.97	0.96	0.92	0.94
	2	0.97	0.92	0.96	0.94
	3	1.00	1.00	1.00	1.00

	class	accuracy	precision	recall_rate	Fmeasure
N5	0	1.00	1.00	0.98	0.99
	1	0.98	0.97	0.96	0.96
	2	0.98	0.95	0.97	0.96
	3	1.00	0.99	1.00	0.99

B. Pose dataset

	class	accuracy	precision	recall_rate
N1	0	1.00	0.99	1.00
	1	0.98	0.95	0.98
	2	0.93	0.97	0.74
	3	0.95	0.83	0.99

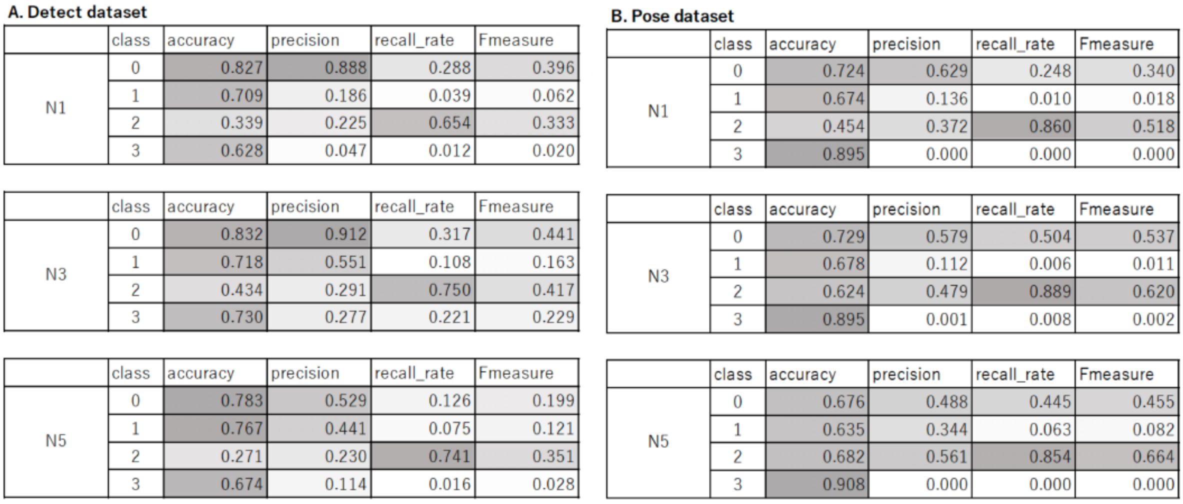
	class	accuracy	precision	recall_rate
N3	0	1.00	0.99	0.99
	1	0.99	0.96	0.99
	2	0.93	0.98	0.73
	3	0.94	0.80	0.99

	class	accuracy	precision	recall_rate
N5	0	0.99	0.98	0.99
	1	0.96	0.92	0.92
	2	0.90	0.89	0.71
	3	0.94	0.81	0.98

FIGURE 3. Evaluation (3 subjects)

5.2. **Evaluation (1 subject).** We conducted an experiment with one subject not used for training. We obtained lower over all results. However, this was expected since "pose data" provided abstracted information of posture, which helps DNN to estimate actions for an individual with unknown body shape in unknown clothes.

6. **Conclusion.** In this research, we propose a method for training movie data by using neural network. Our results showed the best accuracy when we used 5 CNNs and Detect dataset. We were able to estimate human behavior by using only a few frames because the subject was in a confined area not in a spacious area such as lab. In future we aim to improve the versatility of our training data. Currently, we are able to experiment only



A. Detect dataset

	class	accuracy	precision	recall_rate	Fmeasure
N1	0	0.827	0.888	0.288	0.396
	1	0.709	0.186	0.039	0.062
	2	0.339	0.225	0.654	0.333
	3	0.628	0.047	0.012	0.020

	class	accuracy	precision	recall_rate	Fmeasure
N3	0	0.832	0.912	0.317	0.441
	1	0.718	0.551	0.108	0.163
	2	0.434	0.291	0.750	0.417
	3	0.730	0.277	0.221	0.229

	class	accuracy	precision	recall_rate	Fmeasure
N5	0	0.783	0.529	0.126	0.199
	1	0.767	0.441	0.075	0.121
	2	0.271	0.230	0.741	0.351
	3	0.674	0.114	0.016	0.028

B. Pose dataset

	class	accuracy	precision	recall_rate	Fmeasure
N1	0	0.724	0.629	0.248	0.340
	1	0.674	0.136	0.010	0.018
	2	0.454	0.372	0.860	0.518
	3	0.895	0.000	0.000	0.000

	class	accuracy	precision	recall_rate	Fmeasure
N3	0	0.729	0.579	0.504	0.537
	1	0.678	0.112	0.006	0.011
	2	0.624	0.479	0.889	0.620
	3	0.895	0.001	0.008	0.002

	class	accuracy	precision	recall_rate	Fmeasure
N5	0	0.676	0.488	0.445	0.455
	1	0.635	0.344	0.063	0.082
	2	0.682	0.561	0.854	0.664
	3	0.908	0.000	0.000	0.000

Figure 4. Evaluation (1 subject)

from a side angle, but we would like to improve the system so that we can experiment from various angles.

Acknowledgment. This work was partially supported by JSPS KAKENHI Grant-in-Aid for Scientific Research(C) Number 16K00476.

REFERENCES

[1] A. Krizhevsky, I. Sutskever and G. E. Hinton, ImageNet classification with deep convolutional neural networks, *ImageNet Classification with Deep Convolutional Neural Networks*, 2012.

[2] A. L. Caterini and D. E. Chang, Recurrent neural networks, in *SpringerBriefs in Computer Science*, 2018.

[3] S. Ji, W. Xu, M. Yang and K. Yu, 3D convolutional neural networks for human action recognition, *IEEE Trans. Pattern Anal. Mach. Intell.*, 2013.

[4] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia and A. Baskurt, Sequential deep learning for human action recognition, in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2011.

[5] W. Liu et al., SSD: Single shot multibox detector, in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016.

[6] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, You only look once: Unified, real-time object detection, *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp.779-788, 2016.

[7] Q. Chen and V. Koltun, Photographic image synthesis with cascaded refinement networks, *Proc. of the IEEE International Conference on Computer Vision*, pp.1520-1529, 2017.

[8] Z. Cao, T. Simon, S. E. Wei and Y. Sheikh, Realtime multi-person 2D pose estimation using part affinity fields, *Proc. of the 30th IEEE Conf. Comput. Vis. Pattern Recognition (CVPR 2017)*, pp.1302-1310, 2017.

[9] X. Zhou, M. Zhu, S. Leonardos, K. G. Derpanis and K. Daniilidis, Sparseness meets deepness: 3D human pose estimation from monocular video, *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp.4966-4975, 2016.

[10] M. Habermann, W. Xu, M. Zollhöfer, G. Pons-Moll and C. Theobalt, LiveCap: Real-time human performance capture from monocular video, *ACM Trans. Graphics*, vol.38, no.2, pp.1-17, 2019.

[11] L. Sun, K. Jia, D. Y. Yeung and B. E. Shi, Human action recognition using factorized spatio-temporal convolutional networks, *Proc. of the IEEE International Conference on Computer Vision*, 2015.