Quantum Neural Network-Based Deep Learning System to Classify Physical Timeline Acceleration Data of Agricultural Workers

Shinji Kawakura and Ryosuke Shibasaki

Abstract — In this study, we extract agricultural (agri-) workers’ physical acceleration timeline data and execute quantum neural network-based deep learning for data classification. Although various approaches have been implemented globally for indoor and outdoor agri-working sites, there is still scope for improvement. Therefore, in this study, we adapt these approaches mainly for automated, high-tech agri-sites and apply the quantum convolutional neural network (QCNN) deep-learning-based method to qualitatively demonstrate the classification of physical workers’ timeline datasets. For our dataset, our subjects were six experienced and six completely inexperienced male agri-manual workers. The target task was to cultivate a farmland using a simple traditional Japanese hoe in a semi-crouching position. We captured the subjects’ acceleration data using an integrated multi-sensor module mounted on a wooden lilt 150 mm from the gripping position of the dominant hand. We developed and used Python (ver. 3.9) and recent distributed libraries for quantum computation and distributed concerned libraries. For data classification, we successively executed three patterns using QCNN, a category of quantum deep learning (QDL)-based deep learning, and evaluated the systems by obtaining loss, validation accuracy, and final epochs’ elapsed time data. These methods of analyzing digital data are expected to find practical applications and provide key suggestions for improving daily task efficiency.

Index Terms — Quantum Neural Network, TensorFlow Quantum, deep learning-based classification, characteristic extraction, physical timeline data, acceleration.

I. INTRODUCTION

Machine learning-based analysis and categorization methods for matrix-form data have been gaining considerable attention for catering to both societal and academic needs, particularly focusing on agricultural (agri-) workers, managers, technicians, researchers, and nonliving targets (e.g., harvests, and agri-tools).

Although the practical application of various machine learning-based techniques has been improving, deep learning-based methods for analyzing human body part acceleration and angular velocity data are limited, and to the best of our knowledge, promising literature concerning angular velocity data is lacking.

Existing quantum deep learning-based analysis methods in the field of agriculture have focused on targets such as harvests, weeds, forests, and farmers [1]–[3]. However, these studies are insufficient, particularly for the analysis of data related to traditional Japanese workers.

In general, common, non-specific deep learning (convolutional neural network (CNN))-based systems can handle subtleties that a human might fail to notice [4]–[13]. However, in light of past CNN-based industrial and academic achievements, in this study, we base our quantum deep learning-based approach, particularly quantum CNN (QCNN) [14]–[23], a category of Quantum Deep Learning (QDL)-based approach on the achievements in current deep learning fields where timeline physical data are transformed into WAV-format sound file data.

For the QCNN-based method, we select and utilize recent improvements in analyzing agri-work using advanced technologies with various deep learning approaches to understand how those solutions reflect the experience of traditional agri-workers. Furthermore, such approaches can be used to reconstruct data files with missing information.

Specifically, CNN-based achievements in academic and business fields have already reached sufficient levels for real-world application in agriculture fields, providing a large amount of positive results.

However, the application of QCNN-based methods in agriculture fields has not been reported thus far. Hence, in this study, we utilize the characteristics of QCNN that are suitable for the fast analysis of medium- to large-sized sequential digital timeline data.

Based on the aforementioned background and considering future expansions, we developed a numerical QCNN-based deep learning method for data categorization to support agri-informatics researchers, manufacturers, managers, and workers. Further, we evaluated the proposed method for its validity and precision of the quantitative data, and we elucidated the method’s limitations.

Our study results could assist in the development of automatic and high-tech agri systems that will help agri-informatics researchers and manufacturers continually develop diverse techniques to improve agri-worker skills and the utility of existing agri-tools and agri-techniques.

II. METHOD

A. Subject and Target

For the data used in this study, we captured and aggregated original images from non-specific outdoor farmlands for test data. The target subjects were agri-workers with 1-5 years’ experience (Table I).

Published on December 4, 2020.
S. Kawakura, Osaka City University, Japan.
(e-mail: s.kawakura@gmail.com)

DOI: http://dx.doi.org/10.24018/ejfood.2020.2.6.182

R. Shibasaki, The University of Tokyo, Japan.
(e-mail: shiba@cisis.u-tokyo.ac.jp)
Prior to the data collection, we consulted agri-managers and workers to mitigate the difficulties in handling dozens of samples in farmlands. Following consultations with Japanese farmers, we focused on agri-workers using a traditional Japanese hoe, which is the most familiar agri-tool in traditional Japanese small- to medium-sized outdoor farms [1]–[3].

First, we accumulated image datasets related to the task of cultivating a field using a hoe by experienced (aged 25-60 years; n = 5) and completely inexperienced (aged 25-60 years; n = 5) subjects in an outdoor farmland in a semi-crouching posture.

In particular, we considered the vertical acceleration value in the x-direction in Figs. 1-2 and output the accuracy values of the classification of hoe acceleration data from experienced and inexperienced subjects. We captured the subjects’ acceleration data using an integrated multi-sensor module (ATR Promotion’s TSND121) mounted on a wooden lilt 150 mm from the gripping position of the dominant hand.

Fig. 1. Three axes of the acceleration from the sensor module attached to a hoe.

**TABLE I: BASIC INFORMATION RELATED TO SUBJECTS [RANGE, AVERAGE (AVE), STANDARD DEVIATION (SD), AVERAGE Z VALUE (Z), COEFFICIENT OF VARIATION (CV) AND DESCRIPTION]**

<table>
<thead>
<tr>
<th>Index</th>
<th>Experienced Group (n = 6)</th>
<th>Inexperienced Group (n = 6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Experience (year)</td>
<td>(Ave 40.8, SD 14.3, Z =-1.73* 10^-4, CV 2.61)</td>
<td>None</td>
</tr>
<tr>
<td>Age (year)</td>
<td>(Ave 66, SD 8.50, Z 0* 10^-4, CV 34.9)</td>
<td>23-26</td>
</tr>
<tr>
<td>Stature (cm)</td>
<td>(Ave 162.5, SD 5.5, Z 0* 10^-4, CV 3.39)</td>
<td>(Ave 174.8, SD 3.34, Z =-2.85* 10^-4, CV 1.91)</td>
</tr>
<tr>
<td>Weight (kg)</td>
<td>(Ave 69.7, SD 9.96, Z =-1.73* 10^-4, CV 14.3)</td>
<td>(Ave 67.3, SD 7.91, Z 5.92* 10^-2, CV 11.7)</td>
</tr>
<tr>
<td>Dominant hand</td>
<td>Right (n = 5), Left (n = 1) (Some had acute lower back pain)</td>
<td>Right (all subjects)</td>
</tr>
<tr>
<td>Past serious physical disorder</td>
<td>None (None)</td>
<td>None or tennis or badminton (once or twice per week)</td>
</tr>
</tbody>
</table>

Fitness habits: Walking

B. Field

Based on consultations with farmers about future practical applications, particularly in diverse Asian countries, we examined agri-workers using traditional Japanese agri-tools in small- to medium-sized outdoor farms that are not completely trimmed.

C. Data Sensing and Accumulating

Fig. 3 presents the formatted timeline of the processes concerning data sensing and accumulation: (1) time of various initial settings (several dozen minutes), (2) waiting time (10 s), (3) trial time (30 swings with a hoe), (4) waiting time (10 s), and (5) time for various post-handling processes (several minutes).

D. Computing

Based on recent academic trends and past related results in the agri-informatics fields [1]–[6], in this study, we adopt the computing flow consisting of five phases, as shown in Fig. 4.

Fig. 2. Hoe connected to sensors and laptop PC.
(I) In the raw acceleration data gathering phase, we use originally obtained data, not open and distributed datasets owing to the study’s characteristics. For the system, we use distributed API and a program written in Visual Basic language on Visual Studio 2015 to input and save datasets; we obtain the aforementioned original data using the integrated sensor ATR-121 (ATP Promotions Inc., Kyoto, Japan) shown in Figs. 1-2.

(II) We execute the program for transforming the timeline acceleration data into WAV-format sound data (Fig. 5) for the analysis and classification using Python 3.9 with PyTorch 1.7 and other things (e.g., libraries).

For transformed WAV-format data files, we set appropriate lengths and data size (bytes) for successive processing.

![Fig. 5. An example of a transformed WAV-format data from acceleration and angular velocity data.](image)

(III) For the deep learning phase, we compute the QCNN-based method using Google Colaboratory-based open program systems, as done previously [24].

The program is executed using Python 3.9, including PyTorch 1.7, Numpy, Scipy, and Chainer.

The overall process of the system development is as follows:

1. define a method to standardize the data shapes.
2. define the fast Fourier transform (FFT)-based analysis method (in past studies, we have developed deep learning-based systems with hidden layers to perform FFT-based analysis using AI computing, focused on program-based applications).
3. analyze and classify WAV files with the QCNN-based deep-learning method.
4. define and execute the main class.

The steps of the training and validating programs of deep learning are as follows:

(A) For the fundamental setting:
1. import TensorFlow and TensorFlow-Quantum (TFO; the layer-class is designed and constructed considering the practical usage of TensorFlow graph’s circuits).
2. import other necessary factors, such as Cirq distributed by Google Inc., Sympy, Numpy, and Matplotlib-Library.

(B) To set the quantum circuits and input data:
1. define the quantum circuits.
2. transform these circuits into a Tensor object.
3. append the other quantum circuit, change it into an instance, and append the instance to the prepared Tensor object.

4. present and confirm the settled quantum circuits.
5. input the training and validation data.

(C) For the program for defining layers with TensorFlow:
1. define the Cirq-based quantum-framework and define and execute the main class.
2. define the rectangular “cluster state circuits”.
3. define the layers of the model using Cong and “LukinQCNN Paper”.
4. define one-cubit’s unitary-circuit, two-cubits’ unitary-circuit, and two-cubits’ quantum circuit.
5. define one-dimensional quantum-convolution; and
6. pool the quantum-pooling layers from N-cubit to fracN2-cubit using the two-quantum bits’ pool.

(D) For the program for defining the QCNN model with pre-defined layers (Fig. 6):
1. define from eight-qubit, and pool one unit.
2. create one qubit and read out operators in Cirq.
3. build a sequential model enacting the logic; we developers are making the static cluster state prep as a part of the AddCircuit and the “quantum datapoints” appear in the form of executions.
4. display the keras plot of the model.

![Fig. 6. Flowchart concerning Assembling QCNN Circuits in a TensorFlow Flow Graph.](image)

(E) The deep learning model is trained using adequate patches for simplification (Figs. 7, 8).

![Fig. 7. Flowchart of QCNN training.](image)

Figs. 8. Flowchart of cluster state circuits: (1) Circuit for a rectangular cirq.GridQubit, (2) Circuit for one-qubit unitary, and (3) Circuit for two-qubit unitary.
(F) For the first phase (Pattern 1), we use the single QCNN-based deep learning model described below and execute it in a span of 25 epochs (Fig. 9).

1) We insert the aforementioned quantum filter once and execute the deep learning (optimizer = Adam, learning_rate = 0.02, loss = tf (TensorFlow).losses.mse; other variables are as presented in Table II):

![Fig. 9. Architecture of the trial computation of the QCNN-based deep learning model for Pattern 1.](image)

**TABLE II: DATA OF THE DEEP LEARNING-BASED CLASSIFICATION (N = 5)**

<table>
<thead>
<tr>
<th>Item</th>
<th>Pattern 1</th>
<th>Pattern 2</th>
<th>Pattern 3</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Epoch (epoch)</td>
<td>25</td>
<td>25</td>
<td>25</td>
<td>We set these values for all patterns based on past similar studies.</td>
</tr>
<tr>
<td>Batch size</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>Verbose</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Loss</td>
<td>0.42</td>
<td>0.19</td>
<td>0.16</td>
<td></td>
</tr>
<tr>
<td>Validation accuracy</td>
<td>0.88</td>
<td>0.92</td>
<td>0.93</td>
<td></td>
</tr>
<tr>
<td>Final epochs’ elapsed time (ms/sample)</td>
<td>12</td>
<td>26</td>
<td>75</td>
<td></td>
</tr>
</tbody>
</table>

(G) For the second phase (Pattern 2), we use a hybrid model that combines a single QCNN and a single traditional (non-specific) CNN, as described below (Fig. 10).

1) We insert the aforementioned single quantum filter once, calculate all quantum bits, and execute the CNN-based deep learning (optimizer = Adam, learning_rate = 0.02, loss = tf (TensorFlow).losses.mse; other variables are presented in Table II):

![Fig. 10. Architecture of the trial computation of the QCNN-based deep learning model for Pattern 2.](image)

(H) For the third phase (Pattern 3), we use a hybrid model that combines multiple parallel QCNNs (in this case, three parallel QCNNs) and a single traditional (non-specific) CNN, as described below (Fig. 11).

1) We insert the aforementioned single quantum filter once, calculate all quantum bits, and execute the CNN-based deep learning (optimizer = Adam, learning_rate = 0.02, loss = tf (TensorFlow).losses.mse; other variables are presented in Table II):

![Fig. 11. Architecture of the trial computation of the QCNN-based deep learning model for Pattern 3.](image)

![Fig. 12. Graph of deep learning-based classification processes in the three patterns. (The average timeline data of the validation accuracies)](image)

### III. RESULTS

Table II presents the matrix-form numerical data as well as their variables, results, and descriptions for the trials considering past studies utilizing deep learning-based data analyses. The rows under “Loss,” “Validation accuracy,” and “Final epochs’ elapsed time” are all average values. As for “Verbose”, we set 1 for checking the sequential progresses, (#0 = no printout, 1 = progress bar, 2 = step-by-step printout) We set these epochs and elapsed times based on previous studies under similar conditions and performed a numerical comparison. As shown by the graphs in Fig. 12, these graph lines are concerning the average timeline data of “Validation accuracy”. The factors such as speeds of changings, sharpness of these graph lines, and fluctuation of the variables for the three patterns differ after they converge at certain values. The obtained numerical features and variations fundamentally agree with those of previous similar industrial trials and academic studies. As shown in Table II, the **validation accuracy** values are almost similar, whereas the **loss** values differ. The system registered final epochs for patterns 1-3 are 0.42, 0.19, and 0.16, respectively. Further, the final epochs’ elapsed times (ms/sample) for patterns 1-3 are 12 ms, 26 ms, and 75 ms, respectively.
IV. DISCUSSION

Table II presents the quantitative features of the three patterns executed using the proposed models, as described in earlier sections. For the datasets of training data for hoe acceleration, we observed that the trends of loss, validation accuracy, and the final epochs’ elapsed time are comparable to those of the experimental datasets [9], [14]–[20], [22], [24]. Furthermore, we observed that the inestimable qualitative trends of fluctuations (volatilities) of the speed of convergence of the graphs also agree with those of the experimental datasets; the graph of pattern 3 converges the fastest, whereas that of pattern 1 converges the slowest.

However, these results cannot be extrapolated to the deep learning system construction for different combinations of people, tools, and other factors specifically. In addition, we could not definitely describe whether the three patterns were adequate for the evaluation of the systems. Nevertheless, we could present to a certain extent the effectiveness and robustness of the deep learning of the QCNN and QCNN hybrid systems.

V. CONCLUSION AND FUTURE TASKS

In this study, we constructed and demonstrated three cases of QCNN-based deep learning systems for analyzing and classifying timeline acceleration data for the task of cultivating with a hoe in an outdoor farmland. We expect this categorization method to be especially promising for practical application in agri-fields.

In the future, we will verify the accuracy and effectiveness of the proposed system for other computing patterns. Previous studies have presented accuracy-based data to some extent; our future related studies will provide further developments in the categorization and analysis of a variety of agri targets.

Furthermore, considering the background conditions, we will present the results and consultation to agri-system developers for aiding agri-workers and managers. We expect the proposed methods to be widely applied to real agri-working sites to promote productivity and contribute to enhancing the efficiency of manual agri-workers and machines. For instance, new or inexperienced agri-workers can automatically or semi-automatically apply our approaches and enhance their skills in the agriculture field. In the long term, we aim to evaluate and improve the system durability, long-term performance, and other methodological mixing patterns.

ACKNOWLEDGMENT

We extend our heartfelt appreciation to the members of The University of Tokyo, who provided considerable support, feedback, and comments for our research and experiments, and the members of Osaka City University.

REFERENCES


Career: Systems engineering, research for private companies. Development and verification of sensing systems for outdoor agricultural workers.

Dr. Kawakura, Research Center for Artificial Photosynthesis (ReCAP) at Osaka City University/Osaka City, Osaka, Japan. IEEE senior member, Hong Kong Chemical, Biological & Environmental Engineering Society (HKCBEES) senior member.

Ryosuke Shibasaki  Place of birth: Fukuoka Pref. Japan. Date of birth: March 1, 1958. Dr. in Engineering, the University of Tokyo, 1987, Bunkyo-ku, Tokyo, Japan. B.A in Engineering, the University of Tokyo, 1980, Bunkyo-ku, Tokyo, Japan. M.A. in Engineering, the University of Tokyo, 1982, Bunkyo-ku, Tokyo, Japan.

Career: Professor at the Center for Spatial Information Science, University of Tokyo.

Dr. Shibasaki, Center for Spatial Information Science (CSIS), The University of Tokyo, Meguro-ku, Tokyo, Japan, and Department of Socio-Cultural and Socio-Physical Environmental Studies, The University of Tokyo/Kashiwa-shi, Chiba, Japan.