Contents lists available at Science-Gate

International Journal of Advanced and Applied Sciences

Journal homepage: http://www.science-gate.com/IJAAS.html

Utilizing hierarchical extreme learning machine based reinforcement learning for object sorting

Nouar AlDahoul *, ZawZaw Htike

Mechatronics Department, International Islamic University Malaysia, Kuala Lumpur, Malaysia

ARTICLE INFO

Article history:
Received 22 August 2018
Received in revised form
6 December 2018
Accepted 7 December 2018

Keywords:
Object sorting
Reinforcement learning
Hierarchical extreme learning-machine
Deep learning
Feature learning

ABSTRACT

Automatic and intelligent object sorting is an important task that can sort different objects without human intervention, using the robot arm to carry each object from one location to another. These objects vary in colours, shapes, sizes and orientations. Many applications, such as fruit and vegetable grading, flower grading, and biopsy image grading depend on sorting for a structural arrangement. Traditional machine learning methods, with extracting handcrafted features, are used for this task. Sometimes, these features are not discriminative because of the environmental factors, such as light change. In this study, Hierarchical Extreme Learning Machine (HELM) is utilized as an unsupervised feature learning to learn the object observation directly, and HELM was found to be robust against external change. Reinforcement learning (RL) is used to find the optimal sorting policy that maps each object image to the object's location. The reason for utilizing RL is lack of output labels in this automatic task. The learning is done sequentially in many episodes. At each episode, the accuracy of sorting is increased to reach the maximum level at the end of learning. The experimental results demonstrated that the proposed HELM-RL sorting can provide the same accuracy as the labelled supervised HELM method after many episodes.

1. Introduction

Object sorting is one of the most important automatic tasks, with the objective of recognizing different objects varied in colours, sizes, shapes and orientations that map each object to its specific location. Sorting has an important role in the production line, which has attracted many researchers to utilize the vision-based techniques to increase productivity, using the automatic sorting systems (Tho et al., 2016; Tho and Thinh, 2015).

Application of object sorting task is common in agricultural, industrial, and medical sectors. Fruits and vegetables are the examples of objects that need to be sorted and graded in the smart marketing to increase the production. Traditional image processing techniques have been used for grading of fruits into different categories, such as size, shape, colour and texture. Colour-based fruit grading was used to extract colour features to identify the defective fruits from normal ones (Pandey et al., 2013). In the Japanese automobile industry, Japanese cucumbers have been graded by size, shape, colour, and other attributes, using deep learning to sort cucumbers into nine different classes. Sorting and grading of flowers were also applied in the greenhouse and market (Sun et al., 2017) using the multi-input convolutional neural network for the flower sorting. The variable changes in the visual appearance of the fruits and vegetables, as well as the features extracted make the sorting task more challenging (Susnjak et al., 2013). However, many efforts are still being made to improve the accuracies of sorting of fruit varieties.

Object sorting can be done by different machine learning techniques, such as supervised learning and unsupervised learning. In supervised learning, many image samples are labelled manually to perform the classification. Above that, the expert knowledge is required to develop the input/output pairs and this knowledge is not always available. Traditional hand-crafted features depend on colour, length, blob, corner or edge. These methods are application dependent (different features for different applications). Above that, the features are not adaptive to the environmental changes, such as lighting. Features learning take its place as a robust method against external change.

* Corresponding Author.
Email Address: nouar.aldahoul@live.iium.edu.my (N. AlDahoul)

https://doi.org/10.21833/ijaas.2019.01.015
Corresponding author's ORCID profile:
https://orcid.org/0000-0001-5522-0033
2313-626X/© 2018 The Authors. Published by IASE.

Different deep models were used for classification and recognition, and these models require long training because of weights fine-tuning. Graphical Processing Unit (GPU) is used to speed up the learning. Moreover, extreme learning machine with multiple layers has been demonstrated to be fast deep models without weights fine-tuning (Tang et al., 2016). The input weights are generated randomly, the output weights are calculated analytically, and HELM can be run on the Central Processing Unit (CPU). Above that, their performances are comparable with other deep models in the terms of accuracy and learning time (AlDahoul et al., 2018). HELM-RL technique was utilized for maze navigation (Aldahoul et al., 2017), and it was found to outperform gradient based auto-encoder in term of learning time. It also provided a comparable performance with the principal component analysis in term of accuracy.

The objective of this study is to utilize the fast feature learning of HELM in reinforcement learning to find optimal actions after observing high dimensional visual data for objects sorting task. The novelty of this work is as follows:

- This is the first work that utilizes HELM based RL as a fast-deep reinforcement model for object sorting task.
- RL is utilized to learn the optimal behaviour automatically without human intervention (no prior knowledge or labels).
- Reward supervised learning approach is proposed to generate rewards as a replacement of pre-defined reward function.

The paper is structured as follows: In section 2, HELM feature learning, ELM classification, and reinforcement learning methods are summarized. The main steps of the proposed HELM-RL agent are also explained. Section 3 discusses the experimental results and the analysis. The comparison between HELM multi-labelled supervised classification and the proposed HELM-RL is also demonstrated in term of testing accuracy. Section 4 demonstrates the efficiency of the proposed system by summarizing the outcome of this work.

2. Methodology

2.1. Hierarchical ELM for feature learning

Instead of using hand-engineered features, deep models automatically extract hierarchical abstract representations from the data. Hierarchical extreme learning machine is a fast-deep model used to learn features automatically by utilizing unsupervised sparse ELM auto-encoder (Tang et al., 2016). The sparse ELM encoder utilizes the fast-iterative shrinkage-thresholding (FISTA) algorithm, and H-ELM does not require the encoder's weights to be fine-tuned iteratively. This feature assists in reducing the time used for learning/ training significantly. ELM is used in the last layer for classification/regression (Huang et al., 2006), and H-ELM has a good generalization and efficient learning time. Please refer to Tang et al. (2016) for more details concerning H-ELM.

2.2. Reinforcement learning

Reinforcement learning, identified as one of the significant learning methods, focuses on how agents perform optimal actions to get the maximum value of the discounted cumulative reward formulated in Eq. 1 (Sutton and Barto, 2018).

$$R = \sum_{T=0}^{\infty} \gamma^{T}\ r_{T+1} \qquad (1)$$

where 0 <γ<1 represents the discounted factor.

RL framework is represented as a Markov decision process (MDP), which differs from the conventional learning, and it does not require previous information about the environmental model. The basic blocks of the RL model for the sorting task are:

- Environment observations O: images of objects in the start region.
- Agent actions A: selection of orientation and location.
- Reward R: the reward is given to the agent after selecting an action. It is +1 for a positive action and -1 for a negative one.

Q-learning is one of the most common and useful RL algorithms. It is a model-free method. Q-learning depends on updating value function in value iteration algorithm, and its value function is formulated in Eq. 2. The resultant optimal policy is formulated in Eq. 3.

$$Q_f(s,a) = Q_f(s,a) + \ \alpha(R + \ \gamma\, max_{a'}\left(Q_f(s',a')\right) - Q_f(s,a)) \qquad (2)$$

$$\pi\ (s) = arg_a \max \left(Q_f(s,a)\right) \qquad (3)$$

where Q_f represents the value function, α is the rate of learning.

2.3. Classification with extreme learning machine

Extreme learning machine (ELM) is different neural network architecture with a feed forward property, which consists of a single hidden layer. The ability of generalization and efficient learning time are the main reasons to make this method successful (Huang et al., 2006). The weights and biases of the hidden layers are given in a random way. However, the output weights are found analytically.

$$f\ (x) = \sum_{i=1}^{M} F_i(x, W_i\ , b_i).\ \beta_i \qquad (4)$$
$$W_i\ \epsilon\ R^d \ \ ,\ \ b_i\ ,\beta_i\ \ \epsilon\ R$$

where Fi (·) is the activation function of i-th hidden neuron, bi is the bias, Wi is the input weight, βi is the output weight, and M is the nodes number in the hidden layer.

$$\beta = G^{\dagger}T \quad , \beta = G^{T}\left(\frac{1}{\lambda} + G.G^{T}\right)^{-1}.T \qquad (5)$$

where H is the matrix of the hidden layer outputs, T is the target matrix, G† is the Moore–Penrose generalized inverse of G, and λ is the coefficient of regularization.

2.4. HELM-RL agent

The proposed agent has the following steps:

- Parameters Initialization: The weights W0 of the encoder's hidden layers are given random values. The episode counters C and sample counter S are set to zeros. The action value function vector Q (s, a) is also set to zeros.
- Environment exploration: ϵ-greedy policy is used to explore the environment by collecting training samples represented as observations Ot. These observations are entered the H-ELM. S counter is incremented by one after observing a new sample.
- Observation encoding: All training samples are transferred from the space of observations to the space of features by using ELM based auto-encoder to encode the observations and obtain feature vectors Zt, and Zt = Encode (Ot).
- Q-learning based RL: The action value of each feature vector is calculated by utilizing Eq. 4, using supervised ELM as a value function approximator. The 2 to 4 steps are repeated until achieving the convergence.
- Policy testing: The encoder, the approximated action value function and the greedy policy are the outcomes of the previous results. They are used to test the quality of policy with the testing samples.

3. Experiments and results

The experiments are in two stages: training and testing. In the training stage, the inputs of the system are three images. 1) The object in the start area. 2) Empty destination area without an object. 3) Destination area including the object. The latter two images are subtracted from each other to get the difference image, which is used to formulate the reward function. The proposed HELM-RL agent observes the first image at the input, and then selects the action for gripper angle or object's location to finally get a reward. This process is repeated until achieving the optimal performance. In the testing stage, the image of the object in the start area is mapped to the optimal action directly. Fig. 1 illustrates the block diagram of object location/orientation sorting system.

3.1. Gripper angle selection

In the automatic application, a robot gripper is attached to hold an object, and the object is placed in four different orientations (90^{o} - 0^{o} – 45^{o} – 135^{o}). These orientations allow the robot to select an optimal orientation to grip the object and carry it to its specific location in the destination area. A reward is given if the object reaches its destination. If the difference image has black pixels, the grip will not able to hold the object and a negative reward is given. A white region in the difference image refers to an ability of gripper to hold the object and move it to the target area and the positive reward is given. The objective of this task is to accumulate as many rewards as possible to learn how to select the optimal gripper orientation for each object. The task is done by interacting with the environment without human intervention. Fig. 2 shows different objects in four orientations (90^{o} - 0^{o} – 45^{o} – 135^{o}).

3.2. Shapes sorting

Different object shapes are sorted to different locations, and these objects vary in their shapes (rectangle – triangle – x shape). However, there are no input/output pairs available for this task to utilize the supervised learning. RL method is used instead. The objective of this task is to accumulate as many rewards as possible to learn how to map each object observation to the optimal location. The experiment begins by carrying a random object from the start area to a random location in the destination. The difference image gives the shape of the object and the coordinates of its location. The reward function is formulated to give a positive reward if the object is in the right location; however, the negative reward is given otherwise. The reward function is formed as a binary classifier to classify the difference image (black and white) into two classes: good or bad location. Moreover, HELM is used for feature learning and reward learning. Fig. 3 shows different objects in three shapes.

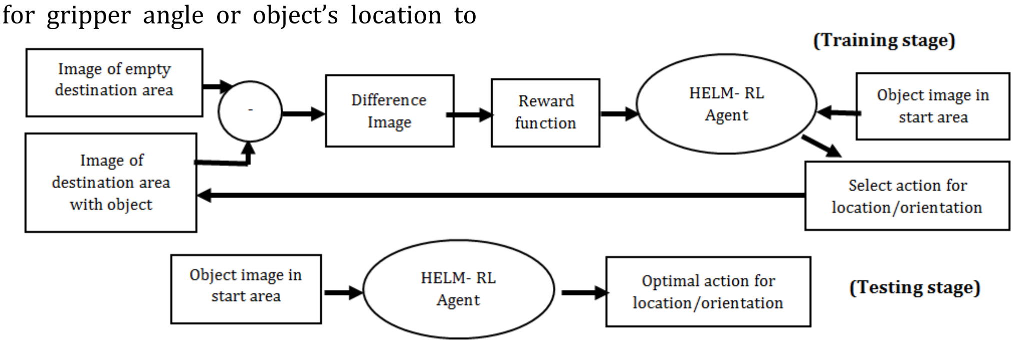
(Training stage)
Image of empty destination area
-
Difference Image
Reward function
HELM- RL Agent
Object image in start area
Image of destination area with object
Select action for location/orientation
Object image in start area
HELM- RL Agent
Optimal action for location/orientation
(Testing stage)

Fig. 1: The block diagram of object location/ orientation sorting system (training and testing stages)

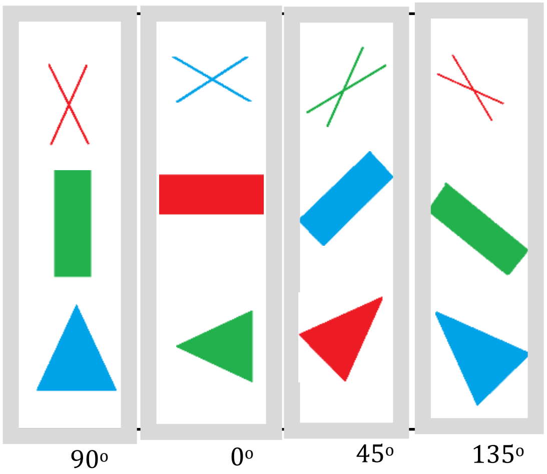

Fig. 2: sorting objects according to their orientations

3.3. Reward function as a binary classifier

Reward function is usually used as a traditional function to give different values in different situations. In the navigation robot task, the reward values are proportional to the distance between the robot and the target and between the robot and the obstacles. In the shape sorting task, the reward values are given to the black and white images that have white shape object and black background. The mapping between these images and the rewards is done using HELM based supervised learning. The HELM is utilized as a binary classifier to give two classes at outputs. These classes are +1 when the object in its correct location and -1 otherwise. The input is the black and white difference image. Fig. 4 shows samples of difference images.

For binary shape sorting, when the objects are white with a black background, two supervised methods may be used: First, the Multi-labelled classifier is used directly to classify the shapes into three classes. The samples should be collected to perform training for the classifier model. Second, binary classifier was proposed which is known to give better accuracy than the multi-classes. Reward function in RL is used as a binary classifier to classify the differences images captured in a controlled environment. This method was utilized to generate the rewards of the MDP model in RL. The two methods: multi-classes and binary are compared in section 3.9.

3.4. Size sorting

The objects are varied in their sizes (large – medium – small). The difference image gives the size of objects by calculating the coordinates of boundary box. The reward function is determined by calculating the area of boundary box of the white object with the black background. After that, it is compared to the threshold to find the size of the object and check its good or bad location status according to the pre-existing if-else rules. Fig. 5 shows different objects in three sizes.

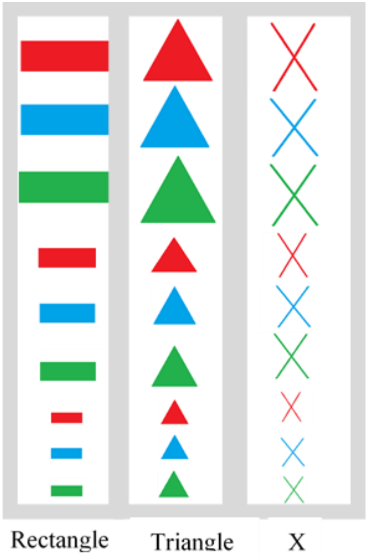

Fig. 3: Sorting objects according to their shapes

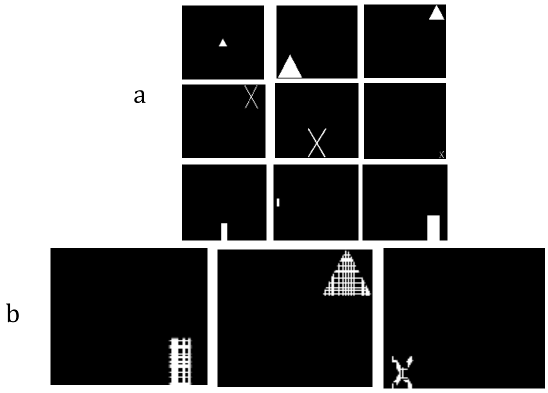

Fig. 4: Samples of difference images; (a) Training samples; (b) Testing samples (with distortion)

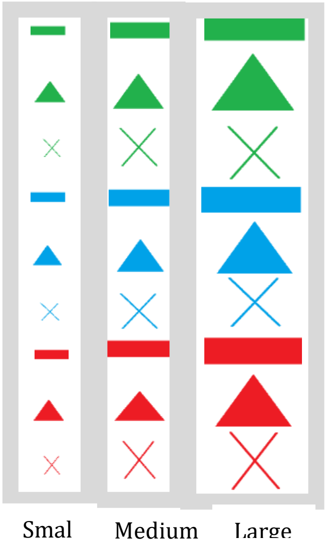

Fig. 5: Sorting objects according to their sizes

3.5. Colour sorting

The objects are varied in their colours (red – green - blue). The difference image after thresholding is multiplied with object image. The resultant image has a black background and coloured object. The reward function is determined by checking the colour of the object to determine its correct or false location according to pre-existing if-else rules. Fig. 6 shows different objects in three colours (red – green - blue).

In the previous sorting tasks, the objects are varied based on their four attributes: orientation, shape, size, and colour. However, each model is learned to sort objects using only one of these attributes. A Q- learning method is used to learn an optimal policy that maps each object observation to a correct location. In the first episode, random objects are put in random locations. After few episodes, the robot arm starts learning the correct mapping. The number of objects in their correct locations is increased with incremental episodes. The accuracy of sorting is given at the end of the learning by dividing the number of object in their correct locations over all objects.

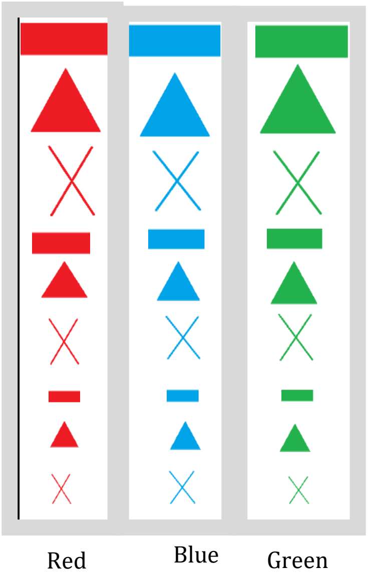

Fig. 6: Sorting objects according to their colours

3.6. Training environment

Two cameras are used in the controlled environment: one is mounted above the start area and another is mounted above the destination area. Destination area is surrounded by a proper environment with a fixed light spot to give a good contrast and eliminate the light change. These environmental settings are only required in the training stage. In the testing stage, the camera mounted at the start area is enough and only input images of the objects are passed to the model to find the correct locations at the output. The object images are coloured and are put in an uncontrolled environment. The environment may have other objects and change in lighting and other external effects. Fig. 7 shows the working field of the sorting system.

3.7. Dataset

A simulated dataset is used to test the proposed method. The number of samples is 972: 500 for training and 472 for testing. These consist of various samples varied as follows:

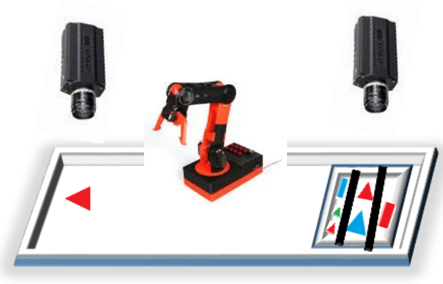

Fig. 7: The working field of the sorting system

1) Four different orientations: angles of 0°, 45°, 90°, 135°.

972 / 4 = 243 samples for each orientation.

2) Three different shapes: rectangle – triangle – x shape.

972 / 3 = 324 samples for each shape.

3) Three different sizes: large – medium – small, and 324 samples for each size.
4) Three different colours: red – green – blue, and 324 samples for each colour.

The k = 10 cross-validation is used to evaluate the results. Random samples are used for training and testing each time. The average accuracy of the 10 is shown in the Tables 1, 2, 3, and 4. Ignoring the colour of the object can increase the accuracy.

3.8. Feature learning

The training is divided into many episodes. Each episode consists of many iterations. In each iteration, a random object image is passed to the HELM based RL model, and HELM is utilized to learn the features of this image in an unsupervised way. This feature vector is mapped to the action value function using ELM.

The features vector F_1 of the observation O is a representation of features vector F after taking the action into consideration by putting F and zeros as follows:

In orientation selection task:

$F_1 = [F\ 0^T\ 0^T\ 0^T]$ when action = 1 angle = 0°
$F_1 = [0^T\ F\ 0^T\ 0^T]$ when action = 2 angle = 45°
$F_1 = [0^T\ 0^T\ F\ 0^T]$ when action = 3 angle = 90°
$F_1 = [0^T\ 0^T\ 0^T\ F]$ when action = 4 angle = 135°
In size, shape, or colour sorting tasks:
$F_2 = [F\ 0^T\ 0^T]$ when action = 1 (size = small, shape = rectangle, colour = red).
$F_2 = [0^T\ F\ 0^T]$ when action = 2 (size = medium, shape = triangle, colour = green).
$F_2 = [0^T\ 0^T\ F]$ when action = 3 (size = large, shape = x shape, colour = blue).

Where 0^T is a row vector of zeros. The number of zeros in 0^T is determined by the number of hidden nodes in ELM based auto-encoder.

F is the feature vector resulted from ELM based auto-encoder.

The ϵ greedy exploration-exploitation is used. The learning rate α = 0.01 and the discount factor γ = 0.95.

3.9. Q-action value function formulation

In this proposed sorting method, there is a small change in the Q-function to fit the task objective, and this change was proposed in Piñol et al. (2015). In Piñol et al. (2015), a new form of Q-function updating equation was proposed with Q-table based RL for object recognition task. In our work, the same idea was applied to the Q-neural network instead of Q-table. In the sorting task, there are no delayed rewards. In other words, only immediate rewards are used to find the action value function. The current state is only affected by its previous visits and not by the next state. The action value function Q_f (s, a) is used as a counter to determine how many times the same feature vectors select the action 'a' in the state's'. This accumulated value is discounted by the factor γ. The new form of the Q value function is formulated by the Eq. 6 as follows:

$$Q_f(s,a) = Q_f(s,a) + \alpha\left(R + \gamma\, max_{a'}\left(Q_f(s,a')\right) - Q_f(s,a)\right) \quad (6)$$

Max (Q_f (s, a')) is used instead of max (Q_f (s', a')). Max (Q_f (s, a')) refers to the maximum action values in the current state. Whereas, max (Q_f (s', a')) represents the maximum action values in the next state. Where s is the current state, and s' is the next state.

Fig. 8 shows the accuracy of reward learner as a binary classifier compared to the traditional multi-labelled classifier. The binary classifier was found to outperform the multi-labelled classifier in the term of accuracy.

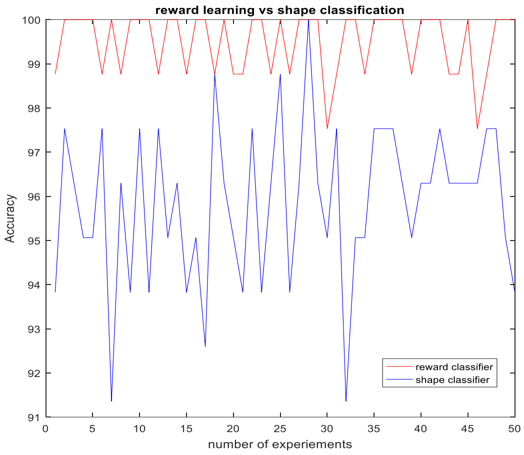

Fig. 8: Reward binary classifier vs multi-classes shape classifier

Figs 9 and 10 show the accuracy of HELM based RL for the shape, size, colour and orientation sorting tasks. The proposed HELM-RL was compared with supervised (multi-labelled) HELM (Tang et al., 2016) in term of accuracy. HELM-RL was found to reach the performance of the supervised HELM (without having training labels) in term of accuracy after many learning episodes.

Table 1 and 2 compare the accuracy of different HELM architectures for a size sorting application. The experiment was repeated 10 times. In each time, random training samples and testing samples were chosen. Then, the average accuracy was calculated. In Table 1, 200 samples were used for training. In Table 2, 500 training samples were used.

Table 1: Average accuracy of size sorting with 200 training samples

Cross validation	HELM 100-1000	HELM 200-2000	HELM 500-5000
1	85.49	84.33	79.27
2	92.49	85.62	84.07
3	88.47	82.51	81.61
4	90.67	86.66	83.42
5	90.16	84.20	84.20
6	95.21	82.77	82.90
7	89.90	82.90	83.42
8	91.97	85.23	84.07
9	89.25	83.68	81.09
10	88.34	82.25	83.68
Average Accuracy	90.20	84.02	82.77

Table 2: Average accuracy of size sorting with 500 training samples

Cross- validation	HELM 100-1000	HELM 200-2000	HELM 500-5000
1	98.52	97.67	94.92
2	96.40	98.31	97.67
3	97.25	98.73	95.34
4	96.40	97.88	94.28
5	96.40	97.46	97.46
6	96.40	95.55	95.76
7	96.40	98.73	97.88
8	98.09	97.25	95.13
9	98.31	98.73	95.76
10	95.34	98.31	95.34
Average Accuracy	96.95	97.86	95.95

Table 3 compares the accuracy of HELM in a shape sorting application with colour and grey images. Converting the colour image to grey increases the accuracy by ignoring the colour attribute of objects sorted according to their shapes. Table 4 compares the accuracy of HELM in orientation sorting applications with colour and grey images, it is also clear that converting the colour image to grey increases the accuracy.

Table 3: Average accuracy of shape sorting with 500 training samples

Cross- validation	HELM With colour	HELM with grey 500-5000	HELM with grey 100-1000
1	68.86	80.51	85.38
2	72.67	85.38	85.38
3	73.09	81.36	82.63
4	70.76	84.75	84.11
5	75.42	80.93	90.25
6	71.82	83.47	91.74
7	69.92	86.02	83.69
8	70.13	84.11	85.38
9	69.70	83.47	80.93
10	69.49	86.65	84.11
Average Accuracy	71.19	83.67	85.36

Table 4: Average accuracy of orientation sorting with 500 training samples

Cross- validation	HELM With colour	HELM with grey 500-5000
1	78.39	87.29
2	81.36	91.10
3	77.75	93.64
4	78.60	87.71
5	77.75	88.56
6	77.12	87.29
7	80.51	92.37
8	77.33	92.37
9	75.64	89.83
10	73.31	89.83
Average Accuracy	77.78	90

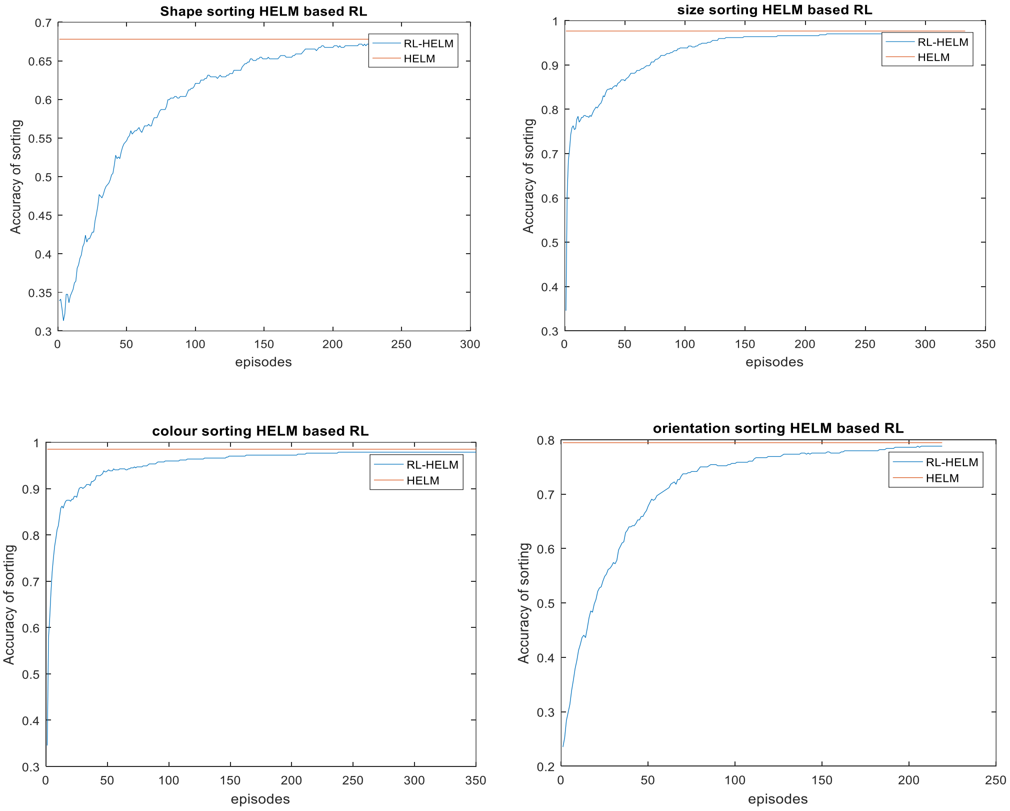

Fig. 9: HELM-RL vs supervised HELM for shape, size, colour, and orientation sorting

4. Discussion and conclusion

In this paper, HELM based RL is proposed to sort objects varied in their colours, sizes, shapes or orientations. Q-learning is used to find the optimal location of the object observation. The advantages of the proposed systems are:

1. There is no need to collect samples before the training. In other words, there is no need to involve human in the training stage. This makes the proposed sorting system automatic and intelligent.
2. The time of feature learning in HELM is short because of the lack of weights tuning stages.
3. The system learns the sorting sequentially and online. The observations are stored in the buffer

in each episode. The model is trained using the batch of samples inside the buffer. This is called experience replay approach.

4. The sorting system can sort new objects in the testing stage by passing the object image directly and getting the correct location in the destination area. This paper uses artificial or simulated samples to prove the efficiency of the proposed system. The future work will utilize real data, such as fruits. The study focuses on sorting the objects according to only one attribute, such as shapes, sizes, colours or orientations each time. The future research needs to use the same model to sort objects according to all attributes at the same time. A new design of MDP is required to achieve this target.

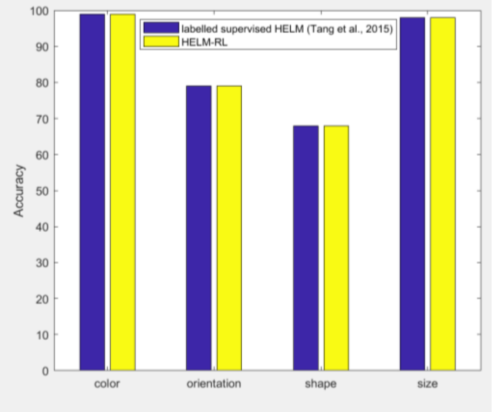

Fig. 10: Comparison between supervised HELM and HELM-RL for different types of sorting

Acknowledgement

This work was supported by the International Islamic University Malaysia under the Research initiatives Grant Scheme (RIGS16-350-0514).

Compliance with ethical standards

Conflict of interest

The authors declare that they have no conflict of interest.

References

AlDahoul N, Htike ZZ, and Akmeliawati R (2017). Hierarchical extreme learning machine based reinforcement learning for goal localization. In the IOP Conference Series: Materials Science and Engineering, IOP Publishing, 184(1): 012055. https://doi.org/10.1088/1757-899X/184/1/012055

AlDahoul N, Sabri M, Qalid A, and Mansoor AM (2018). Real-time human detection for aerial captured video sequences via deep models. Computational Intelligence and Neuroscience, 2018: Article ID 1639561, 14 pages. https://doi.org/10.1155/2018/1639561

Huang GB, Zhu QY, and Siew CK (2006). Extreme learning machine: Theory and applications. Neurocomputing, 70(1-3): 489-501. https://doi.org/10.1016/j.neucom.2005.12.126

Pandey R, Naik S, and Marfatia R (2013). Image processing and machine learning for automated fruit grading system: A technical review. International Journal of Computer Applications, 81(16): 29-39.

Piñol M, Sappa AD, and Toledo R (2015). Adaptive feature descriptor selection based on a multi-table reinforcement learning strategy. Neurocomputing, 150: 106-115. https://doi.org/10.1016/j.neucom.2014.03.080

Sun Y, Zhu L, Wang G, and Zhao F (2017). Multi-input convolutional neural network for flower grading. Journal of Electrical and Computer Engineering, 2017: Article ID 9240407, 8 pages. https://doi.org/10.1155/2017/9240407

Susnjak T, Barczak A, and Reyes N (2013). A decomposition machine-learning strategy for automated fruit grading. In the World Congress on Engineering and Computer Science, San Francisco, USA, 2: 819-825.

Sutton RS and Barto AG (2018). Reinforcement learning: An introduction. MIT Press, Cambridge, USA.

Tang J, Deng C, and Huang GB (2016). Extreme learning machine for multilayer perceptron. IEEE Transactions on Neural Networks and Learning Systems, 27(4): 809-821. https://doi.org/10.1109/TNNLS.2015.2424995 **PMid:25966483**

Tho TP and Thinh NT (2015). Design and development of the sorting system based on robot. In the 15th International Conference on Control, Automation and Systems, IEEE, Busan, South Korea: 1639-1644. https://doi.org/10.1109/ICCAS.2015.7364620

Tho TP, Thinh NT, and Bich NH (2016). Design and development of the vision sorting system. In the International Conference on Green Technology and Sustainable Development, IEEE, Kaohsiung, Taiwan: 217-223. https://doi.org/10.1109/GTSD.2016.57