Human-in-the-loop Reinforcement Learning for Emotion Recognition

Swee Yang Tan
Lee Kong Chian Faculty of Engineering and Science
Universiti Tunku Abdul Rahman (UTAR)
Selangor, Malaysia
tanryan001@1utar.my

Kok-Lim Alvin Yau
Lee Kong Chian Faculty of Engineering and Science
Universiti Tunku Abdul Rahman (UTAR)
Selangor, Malaysia
yaukl@utar.edu.my

***Abstract*—Facial emotion recognition (FER) analyses a person's facial expressions in images to understand the person's emotions. Achieving high accuracy in real-world FER systems requires extensive data, which poses challenges in cost, practicability, and privacy, especially in ensuring fairness and avoiding discrimination related to skin color or health conditions. This paper introduces a novel approach called two-state Q-learning with human feedback (TS-QL-HF) to enhance the accuracy of a FER system by integrating human feedback collected during the evaluation process as a refined reward function into two-state Q-learning (TS-QL) and double Q-learning (DQL) to operate in deterministic environments. This approach does not need an extensive dataset to include minority populations. Simulation results demonstrate that TS-QL-HF provides the highest accuracy. However, the tuned TS-QL, without human feedback, shows a higher efficiency despite a slightly lower accuracy due to the inaccurate reward function.**

Keywords—reinforcement learning, human-in-the-loop, augmented intelligence, emotion recognition

I. Introduction

Facial emotion recognition (FER) refers to a computer's capability to understand and differentiate human facial emotions. Reinforcement learning (RL) enables agents to learn optimal policies for Markovian tasks through reward-based trial-and-error interactions with the operating environment [1]. Recent studies [2]–[4] have demonstrated the effectiveness of RL in enhancing computer vision tasks. Q-learning (QL) is a popular RL algorithm. In QL, an agent (or a decision maker) observes the state (or decision-making factors) and selects an action based on its policy at decision epoch t. The agent observes the next state and delayed reward, which represent the consequences of the state-action pair and updates the Q-value of the state-action pair at the next decision epoch $t+1$. Q-value represents the long-term reward of a state-action pair, so it represents the appropriateness of selecting the action under the state. The policy is a collection of state-action pairs and their respective Q-values. The success of an RL application requires a well-designed function for the delayed reward, which is often challenging in reality [5]. This challenge can be tackled by using human feedback to adjust the delayed reward. For achieving an optimal policy, the agent must achieve the right balance between exploration and exploitation by adjusting its hyperparameters, including the learning rate α and discount factor γ, under deterministic and stochastic environments [1], [5]. The consequences of a state-action pair are consistent (inconsistent) and predictable (unpredictable) in the deterministic (stochastic) environment.

This study focuses on improving and tailoring the two-state QL (TS-QL) approach [2], which is an enhanced variant of the traditional QL approach for a deterministic environment, where the outcome of every action (image transformation) consistently produces the same results when performed under identical conditions and hyperparameters. The goal of TS-QL is to identify the optimal action that maximizes the standard deviation of the feature map in misclassified images by performing image transformations, such as rotation. By doing so, this technique enhances the visual quality of the feature map, thereby improving the clarity of the facial emotion features captured within images. TS-QL has been shown to improve the accuracy of FER systems [4]. Also, TS-QL reduces the complexity of managing vast state-action spaces, which is a common challenge faced by QL approaches, by utilizing only two states, which represent positive and negative outcomes. Double QL (DQL) is an extension of the traditional QL approach that addresses the overestimation of Q-values.

The key contributions of this research are to: a) design TS-QL-HF, which incorporates human feedback as an alternative reward function for TS-QL, for enhancing the quality of rewards; b) develop an action selection strategy and fine-tune hyperparameters for improving the convergence rate while operating in deterministic environments; c) separate the learning of unrelated policies among multiple agents for improving convergence; and d) investigate the use of DQL in deterministic environments. The rest of this paper is organized as follows: Section II presents related work. Section III presents the system model. Section IV presents the RL algorithms. Section V presents simulation results and discussion. Finally, Section VI presents the conclusion of the study and future works.

II. Related Works

This section presents existing literature on human-in-the-loop in real-life FER systems, the exploration strategies, and DQL for addressing the overestimation of Q-values.

A. Human-in-the-loop in real-life FER systems

Societal norms and cultural differences are found to influence the level of expression of facial emotions, causing bias in algorithms [6]. Pure data-driven FER, which learns purely based on data, may be inaccurate in real-life situations due to insufficient data on minority populations [6]. This is because widely representative data can be expensive, difficult, and impractical to collect. This leads to discrimination based on skin color, ethnic origin, or medical conditions related to the face. For instance, a study [7] shows that negative emotions are often misassigned to African descent faces. To improve the accuracy of FER without collecting a tremendous amount of data, human intelligence is integrated into the FER system, improving its reliability and robustness [8]. In this study, human intelligence is utilized to evaluate the correctness of image reclassification after image transformation is applied. Model-based human-in-the-loop algorithms are suitable for FER, where it is difficult to model the reward function and transition from the environment and human [9]. So, reward is used in this study represent the correctness of reclassification, which are either positive or negative.

B. *Exploration Strategies*

Action selection strategies enable an agent to achieve the right balance between exploration and exploitation. Action selection strategies [11], such as ε-greedy and softmax, are highly effective, but they are not suitable for a fully deterministic environment. In the deterministic environment, it is ideal for the exploration to stop once the Q-values converge as further exploration is redundant. Reward-based epsilon decay (RBED) [10] introduces a dynamic exploration rate that reduces the exploration probability ε according to its current reward. Despite the improvement, the agent still suffers from a lower convergence rate due to unnecessary exploration. This study designs an exploration strategy called one-shot RBED for FER.

C. *Double Q-learning for Addressing the Overestimation of Q-values*

DQL [11] is an extension of the traditional QL approach that addresses the overestimation of Q-values caused by the max operator used in the update learned function of traditional QL. The overestimation means that the learned Q-values appear to be higher than their true values [11], which leads to a slower convergence in complex tasks [11], [12]. To solve the problem of the overestimation of Q-values, DQL introduces the use of double estimators instead of a single estimator during learning, which means DQL uses two separate Q-tables that correspond to two update functions. This method is proven to not overestimate, thus making the Q-values more accurate. However, sometimes it might suffer from the underestimation of Q-values [11]. References [11], [12] focuses on investigating DQL in a stochastic environment, where there are uncertainties in the consequences of actions. Therefore, the effects of using DQL in a fully deterministic environment remains unknown. This study investigates DQL and evaluates its effectiveness in improving the convergence rate under such an environment.

III. System Model

The FER system recognizes human facial emotions from images using convolutional neural network (CNN) and RL as shown in Fig. 1. The image loading and preprocessing module performs three main tasks on raw and unstructured images: a) resizing image size to 75×75 pixels; b) removing the fourth dimension (i.e., the transparency layer); and c) normalizing the intensity values of the image into the [0,1] range. The outputs are structured images, which are fed into CNN, helping to improve the convergence rate.

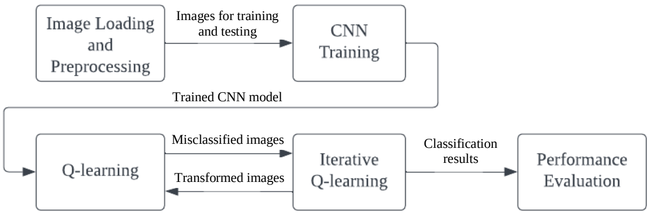


Fig. 1. The FER system.

The CNN module classifies an image into one of the seven human emotion classes, namely anger, disgust, fear, joy, neutral, sadness, and surprise. CNN, which is based on joint feature and classifier learning [13], has been shown to achieve a higher accuracy on large datasets in image classification. The selected CNN model is InceptionV3 [14], which has been shown to outperform other models (e.g., ResNet50 [15] and MobileNetV2 [16]) in image classification. InceptionV3 has been pre-trained and custom classifier layers. During training, the pre-trained layers are fixed, and the custom classifier layers are adjusted. The custom classifier layers consist of four layers, including: a) flatten converts signals from pre-trained layers into long continuous linear vectors; b) dense classifies signals into 1024 categories; c) dropout prevents overfitting; and d) another dense classifies signals into the seven human emotion classes. The InceptionV3 model has 192 layers, 23.9 million parameters (including weights and biases), and 2.1 million trainable parameters in the custom classifier layers.

The QL and iterative QL modules learn the optimal image transformations for all misclassified images in the testing dataset and replace them with transformed images. The iterative QL module uses: a) TS-QL; b) TS-QL-HF, which uses traditional QL; c) TS-DQL; and d) TS-DQL-HF, which uses DQL. The states represent the positive and negative rewards. The actions are to rotate misclassified images by 90 or 180 degrees, or to perform diagonal translations with 15 pixels to the right or to the bottom. The traditional QL approach uses a binary environment reward function, where m and m_1 are the standard deviation of the feature map before and after an image transformation. The following are the reward function r_t and state function s_t:

$$r_t = \begin{cases} 1, & if\ m_1 > m \\ -1, & otherwise \end{cases} \tag{1}$$

$$s_t = \begin{cases} 0, & if\ r_t = 1 \\ 1, & otherwise \end{cases} \tag{2}$$

where state $s_t = 0$ and $s_t = 1$ represent a positive and negative reward, respectively. QL uses Q-function to update Q-values, which are stored in the Q-table as follows:

$$Q_{t+1}(s_t,a_t) = Q_t(s_t,a_t) + \alpha_t(s_t,a_t)\left[r_t + \gamma \max_a Q_t(s_{t+1},a_{t+1}) - Q_t(s_t,a_t)\right] \quad (3)$$

DQL uses two Q-functions to update Q-tables *A* and *B*, respectively. The a^* and b^* parameters represent the current best possible actions in Q-tables *A* and *B*, respectively, as follows:

$$Q^A_{t+1}(s_t,a_t) = Q^A_t(s_t,a_t) + \alpha(s_t,a_t)\big(r_t + \gamma Q^B_t(s_{t+1},a^*_t) - Q^A_t(s_t,a_t)\big)(4)$$

$$Q^B_{t+1}(s_t,a_t) = Q^B_t(s_t,a_t) + \alpha(s_t,a_t)\big(r_t + \gamma Q^A_t(s_{t+1},b^*_t) - Q^B_t(s_t,a_t)\big)(5)$$

When the DQL process has completed, the average Q-values from both tables is calculated, and the average Q-table is used for selecting the final optimal action.

$$\overline{Q_t(s_t,a_t)} = \frac{Q^A_t(s_t,a_t) + Q^B_t(s_t,a_t)}{2} \quad (6)$$

The QL selects the final optimal action by getting the maximum Q-value in the Q-table while DQL selects it by getting the maximum Q-value in the average Q-table.

IV. Reinforcement Learning Approaches

This section explains the RL approaches.

A. Two-State QL (TS-QL)

TS-QL is a revised QL approach for a fully deterministic environment. The enhancements include: a) one-shot RBED, which is a novel action selection strategy, is a variant of ε-greedy that decays ε toward 0 once a positive reward is received, thus stopping exploration as further exploration does not provide any accuracy improvement; and b) the learning of unrelated policies is separated among multiple agents, preventing divergence in learning a new unrelated policy, where the optimal image transformation action of each misclassified image does not relate to others. In addition, the hyperparameters are revised to $\alpha = 1$ and $\gamma = 0$ which are optimal for a fully deterministic environment. TS-QL is efficient as it does not require human inputs, which makes it suitable for situations in which collecting human inputs is impractical. However, TS-QL provides a lower accuracy enhancement to the FER system as the reward function is less accurate. Algorithm 1 is self-explanatory. For simplicity, time instants which are self-explanatory are not shown. Algorithm 1 shows TS-QL. The Q-table is initialized to zeros for all state-action pairs and ε is set to 1. Actions are selected using ε-greedy. The feature map and its standard deviation m of a misclassified image are obtained from the output layer of the CNN model. Action is selected for the misclassified image. The feature map and its standard deviation m_1 of a misclassified image are obtained from the output layer of the updated CNN model. The reward is calculated, and if it is $r = 1$, the next state is set to 0 and ε is decayed toward 0; otherwise, the next state is set to 1. The Q-value $Q(s,a)$is updated using Eq. (3). The episodes repeat until the Q-table finalizes.

Algorithm 1: TS-QL

```
input: misclassified image
output: $Q$
$Q$ ← initialize $Q$ for $\forall_s \in S$ and $\forall_a \in A_s$ with 0, $\varepsilon = 1$
for each episode do
    $a$ ← select action using $\varepsilon$-greedy with $\varepsilon$ value
    feature maps ← get the feature map in the output layer of a CNN model using misclassified image
    $m$ ← calculate the standard deviation of the feature map
    modified image ← apply action on misclassified image
    feature maps ← get the feature map in the output layer of a CNN model using modified image
    $m_1$ ← calculate the standard deviation of the feature map
    $r$ ← get $r$ using environmental reward function
    if ($r$ = 1)
        $s_{t+1} = 0$
        decay $\varepsilon$ toward 0
    else
        $s_{t+1} = 1$
    $s_t \leftarrow s_{t+1}$
    update $Q(s,a)$ using Eq. (3)
end for
$Q$ ← get $Q$ after completing the episodes
```

B. Two-State QL with Human Feedback (TS-QL-HF)

Instead of using standard deviation, TS-QL-HF uses human feedback of 1 or -1, which represent correct and incorrect reclassification, respectively, as the reward function as shown in Step 4 of Algorithm 2.

Algorithm 2: TS-QL-HF

```
    input: misclassified image
    output: $Q$
1   $Q$ ← initialize $Q$ for $\forall_s \in S$ and $\forall_a \in A_s$ with 0, $\varepsilon = 1$
2   for each episode do
3       $a$ ← select action using $\varepsilon$-greedy with $\varepsilon$ value
4       $r$ ← gets $r$ from human evaluator
5       if ($r$ = 1)
6           $s_{t+1} = 0$
7           decay $\varepsilon$ toward 0
8       else
9           $s_{t+1} = 1$
10      $s_t \leftarrow s_{t+1}$
11      update $Q(s,a)$ using Eq. (3)
12  end for
13  $Q$ ← get $Q$ after completing the episodes
```

C. Two-State Double QL (TS-DQL)

TS-DQL uses two separate Q-tables and updates their respective Q-functions, namely Q^A and Q^B using Eqs. (4) and (5), respectively. The output of TS-DQL $\bar{Q}$ is the average of Q^A and Q^B. Hence, TS-DQL may slow down the convergence rate while not improving accuracy since, in a fully deterministic environment, the overestimation of Q-values may not occur [11].

Algorithm 3: TS-DQL

```
input: misclassified image
output: $\bar{Q}$
Q-table ← initialize $Q^A$, $Q^B$for $\forall_s \in S$ and $\forall_a \in A_s$ with $0, \varepsilon = 1$
for each episode do
    $a$ ← select action using $\varepsilon$-greedy with $\varepsilon$ value
    feature maps ← get the feature map in the output layer of a CNN model using misclassified image
```

```
        $m$ ← calculate the standard deviation of the feature maps
        modified image ← apply action on misclassified image
        feature maps ← get the feature map in the output layer of a CNN model using modified image
        $m_1$ ← calculate the standard deviation of the feature maps
        $r$ ← gets $r$ using environmental reward function
        if ($r$ = 1)
            $s_t = 0$
            decay $\varepsilon$ toward 0
        else
            $s_t = 1$
        $s_t \leftarrow s_{t+1}$
        if update $A$
            update $Q^A$ using Eq. (4)
        else if update $B$
            update $Q^B$ using Eq. (5)
        end if
    end for
    $\bar{Q}$ ← calculate the mean of $Q^A$ and $Q^B$ using Eq. (6)
```

D. Two-State Double QL with Human Feedback (TS-DQL-HF)

TS-DQL-HF has the same process as TS-QL-HF except using two q-tables and update functions as mentioned above. TS-DQL-HF is also not recommended as it not only slows down the convergence rate, but further reduces the accuracy enhancement provided by TS-QL-HF.

Algorithm 4: TS-DQL-HF

```
    input: misclassified image
    output: $\bar{Q}$
    Q-table ← initialize $Q^A$, $Q^B$ for $\forall_s \in S$ and $\forall_a \in A_s$ with 0, $\varepsilon = 1$
    for each episode do
        $a$ ← select action using $\varepsilon$-greedy with $\varepsilon$ value
        $r$ ← gets $r$ from human evaluator
        if ($r$ = 1)
            $s_t = 0$
            decay $\varepsilon$ toward 0
        else
            $s_t = 1$
        $s_t \leftarrow s_{t+1}$
        if update $A$
            update $Q^A$ using Eq. (4)
        else if update $B$
            update $Q^B$ using Eq. (5)
        end if
    end for
    $\bar{Q}$ ← calculate the mean of $Q^A$ and $Q^B$ using Eq. (6)
```

V. Simulation Results and Discussion

This section presents simulations to compare the performance achieved by TS-QL, TS-QL-HF, TS-DQL, and TS-DQL-HF. Ethical approval has been granted for this project by Universiti Tunku Abdul Rahman, and informed consent was obtained from all human players. During the simulation, an input is requested from the human evaluator using a popup dialogue box in each decision epoch, and the input r is used in the update of the Q-values (see Algorithm 4).

A. Simulation Setup and Parameters

1) Dataset: The FERG dataset [17] is a database comprised of six cartoon characters, three males (Ray, Malcolm, and Jules) and three females (Bonnie, Mery, and Aia). The dataset contains 55,769 images of facial expressions labeled with seven human emotion classes, namely anger, disgust, fear, joy, neutral, sadness, and surprise, as shown in Fig. 2. The images are consistent and well controlled, so all faces are front facing and not obstructed. The dataset is split into 90% training and 10% test sets, and the training set is further split into 80% for training and 20% for validation.

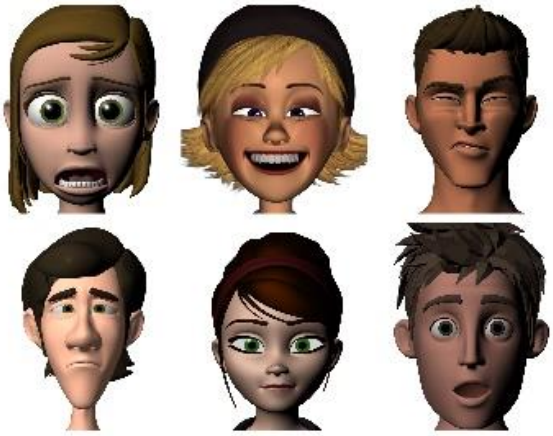


Fig. 2. Sample images from the FERG dataset. From left to right in the top row are Aia with the *fear* expression, Bonnie with *joy*, and Jules with *disgust*. From left to right in the bottom row are Malcolm with *sadness*, Mery with *neutral*, and Ray with *surprise* [17].

2) Parameters and Values: Hyperparameters are selected based on [4]. There are three hyperparemeters used in CNN models as shown in Table I. First, the optimizer are used to update weights and biases of a model to reduce its error. Second, the learning rate α determines the step size of each iteration while minimizing the loss function. Third, the batch size determines the number of samples propagated from the input layer through the hidden layers to the output layer in each iteration before updating the model parameters.

TABLE I. Hyperparameters of CNN Models

Parameters	**Values**
Optimizer	Adam
Learning rate α	0.001
Batch size	20

There are four hyperparameters in QL as shown in Table II. First, the learning rate α determines the importance of past knowledge. Second, the discount factor γ determines the importance of future rewards. In a fully deterministic environment, $\alpha = 1$ and $\gamma = 0$ are optimal [1]. This is because the past information and future reward in this environment provides no additional useful information since their outcomes are already known [1]. Third, the Q-values are initialized to the 1 value as random values can slow down learning. Fourth, the number of episodes is 10, which is sufficient for the small 2×3 state-action space.

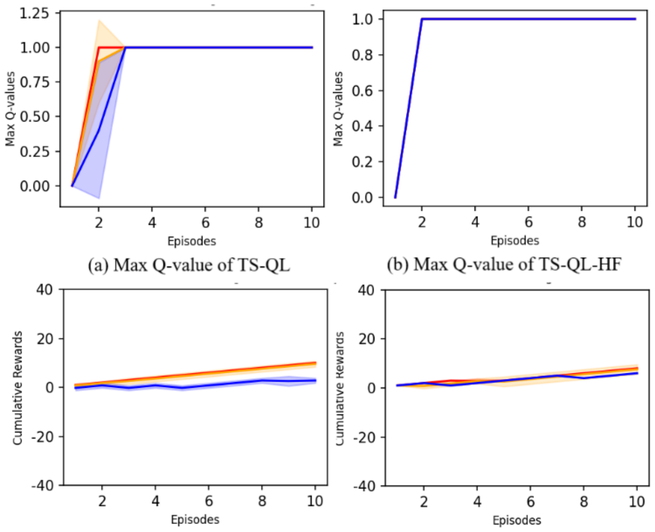
TABLE II. HYPERPARAMETERS OF QL MODELS

Parameters	Values
Learning rate α	1.0
Discount factor γ	0.0
Q-values initialization	1.0
Number of episodes	10

3) Performance Metrics: There are four performance metrics. First, *accuracy* measures the number of correct predictions. Second, the *F1 score* is the harmonic means of precision and recall, thus, it symmetrically represents both precision and recall in a single metric. Third, the *maximum (max) Q-value* represents the Q-value of the optimal action under a particular state. Fourth, the *cumulative reward* represents the amount of rewards which has been received by the agent up to the current training episode. The ordinance of the graphs is the training episode.

B. Experiment 1: Comparison between TS-QL and TS-QL-HF

This section investigates the effects of action selection strategies in TS-QL and TS-QL-HF, and the effects of human feedback in TS-QL-HF compared to TS-QL. Fig. 3 compares the max Q-value and cumulative reward of the three action selection strategies using TS-QL and TS-QL-HF. Table III compares the accuracy and F1 score of the baseline approach, TS-QL and TS-QL-HF with one-shot RBED. The baseline approach uses the InceptionV3 model without the RL approach. The three action selection strategies do not affect the accuracy and F1 score for both TS-QL and TS-QL-HF.



(a) Max Q-value of TS-QL (b) Max Q-value of TS-QL-HF

(c) Cumulative reward of TS-QL (d) Cumulative reward of TS-QL-HF

Fig. 3. Comparison of max Q-values for the three action selection strategies using TS-QL in (a) and TS-QL-HF in (b), and cumulative reward for the three action selection strategies using TS-QL in (c) and TS-QL-HF in (d). *Red* represents one-shot RBED, *yellow* represents harmonic RBED, and *blue* represents random selection. The shaded area represents the standard deviation, which is the achievable range.

TABLE III. ACCURACY AND F1 SCORE IN COMPARISON BETWEEN TS-QL AND TS-QL-HF

	Accuracy	F1 Score
Baseline	0.9849 ± 0.0168	0.9826 ± 0.0199
TS-QL-HF with one-shot RBED	**0.9936 ± 0.0137**	**0.9933 ± 0.0140**
TS-QL-HF Harmonic RBED	**0.9936 ± 0.0137**	**0.9933 ± 0.0140**
TS-QL-HF Random Selection	**0.9936 ± 0.0137**	**0.9933 ± 0.0140**
TS-QL One-Shot RBED	0.9893 ± 0.0143	0.9882 ± 0.0165
TS-QL Harmonic RBED	0.9893 ± 0.0143	0.9882 ± 0.0165
TS-QL Random Selection	0.9893 ± 0.0143	0.9882 ± 0.0165

Fig. 3(a) shows that the maximum (max) Q-value of one-shot RBED (red) converges first, followed by harmonic RBED (yellow) and random selection (blue) for TS-QL. This is attributed to: a) the more conservative exploration approach of the harmonic RBED approach that decays the *ε* using the harmonic sequence; and b) the selected actions of the random selection approach that are random in nature without taking rewards into consideration, causing unimportant parts of the action space being explored. Fig. 3(c) shows that the cumulative reward of one-shot (red) and harmonic (yellow) RBEDs are similar, and they are higher than that of random selection (blue). This suggests that one-shot RBED achieves the best possible convergence rate and cumulative reward, and further explorations after a positive reward is received do not improve the accuracy.

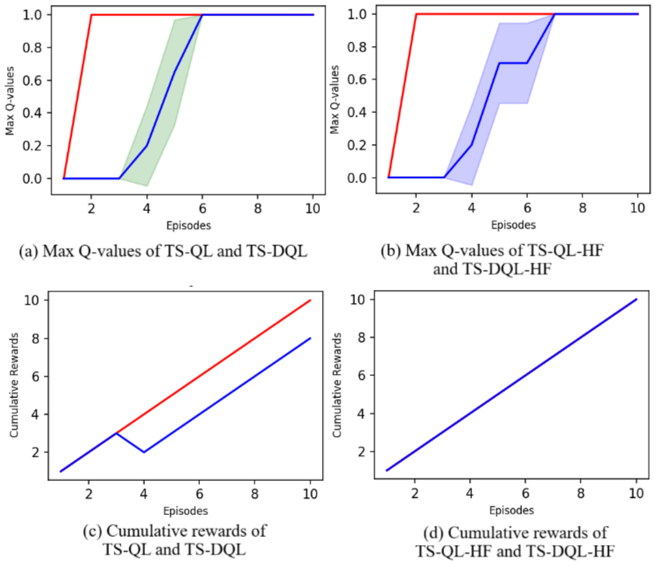
Fig. 3(b) shows that the max Q-values of TS-QL-HF has a similar trend with that in Fig. 3(a). Fig. 3(d) shows that the cumulative reward of TS-QL-HF is lower than that of TS-QL shown in Fig. 3(c); however, the accuracy and F1 score are higher than that of TS-QL as shown in Table III. This is attributed to human feedback, which provides a more accurate reward function compared to the environmental reward, which is given by the standard deviation of the feature map and the variation of feature map intensities. The standard deviation measures the variability of the spread of intensity values in the feature map. A higher standard deviation indicates that the features captured are more distinct and clearer, which leads to improved emotion feature representations. However, it is worth noting that TS-QL-HF is more costly, particularly in terms of time, due to the time spent in collecting human feedback.

In short, the one-shot RBED approach has been shown to be the best possible action selection strategy for the FER system due to its efficiency in exploiting the learned knowledge, leading to a higher cumulative reward. The TS-QL-HF approach has been shown to achieve the best possible prediction accuracy and F1 score for FER.

C. Experiment 2: Comparison between QL and DQL

This section replaces QL with DQL to investigate the effects of using TS-DQL and TS-DQL-HF. Fig. 4 compares the max Q-value and cumulative reward of TS-QL and TS-DQL, and TS-QL-HF and TS-DQL-HF, respectively. Table IV compares the accuracy and F1 score of the baseline approach, TS-QL, TS-QL-HF, TS-DQL, and TS-DQL-HF.

Fig. 4(a) shows that TS-QL (red) converges first, followed by TS-DQL (blue). This is attributed to the more conservative exploration approach of the TSDQL, causing a lower cumulative reward as shown in Fig. 4(c). However, both TS-QL and TS-DQL achieve the same accuracy and F1 score. Similar trends are observed for the comparison between TS-QL-HF and TS-DQL-HF as shown in Fig. 4(b), causing a slightly lower cumulative reward as shown in Fig. 4(d). TS-DQL-HF also achieves lower accuracy and F1 score compared to TS-QL-HF. The traditional QL approach is preferred over DQL in this FER system.



(a) Max Q-values of TS-QL and TS-DQL

(b) Max Q-values of TS-QL-HF and TS-DQL-HF

(c) Cumulative rewards of TS-QL and TS-DQL

(d) Cumulative rewards of TS-QL-HF and TS-DQL-HF

Fig. 4. Comparison of max Q-values between TS-QL (red) and TS-DQL (blue) in (a), and between TS-QL-HF and TS-DQL-HF in (b), and cumulative reward between TS-QL and TS-DQL in (c), and cumulative reward between TS-QL-HF and TS-DQL-HF in (d).

TABLE IV. ACCURACY AND F1 SCORE IN COMPARISON BETWEEN QL AND DOUBLE QL

	Accuracy	**F1 Score**
Baseline	0.9849 ± 0.0168	0.9826 ± 0.0199
TS-DQL-HF	0.9914 ± 0.0142	0.9904 ± 0.0164
TS-QL-HF	**0.9936 ± 0.0138**	**0.9925 ± 0.0162**
TS-DQL	0.9893 ± 0.0143	0.9882 ± 0.0165
TS-QL	0.9893 ± 0.0143	0.9882 ± 0.0165

VI. CONCLUSION

FER systems should be fair for everyone regardless of their skin colors, ethnicities, or health conditions, to avoid any discriminations. This study has proven the effectiveness of using human intelligence in tackling this issue without needing a huge amount of data related to the minority population, which can be difficult and impractical to obtain. This study has also proven the effectiveness of optimal hyperparameters, single-objective agent learning, and one-shot RBED in improving the convergence rate for QL in a fully deterministic environment. Additionally, this study has also proven the ineffectiveness of DQL in a fully deterministic environment. This study suggests a further study to investigate the effectiveness of the continuous improvement in the robustness of the model in evaluating minority populations. This study also suggests a further study to combine TS-QL and TS-QL-HF into a hybrid version to achieve a balance between the efficiency and accuracy offered by both approaches.

ACKNOWLEDGMENT

The research was supported by the Ministry of Higher Education (MOHE), Malaysia through the Fundamental Research Grant Scheme (FRGS/1/2022/ICT02/UTAR/01/2).

REFERENCES

[1] R. Sutton and A. Barto. 2018. *Reinforcement Learning: An Introduction.* MIT Press.

[2] A. M. Hafiz, "Image classification by reinforcement learning with two-state Q-learning," in Handbook of Intelligent Computing and Optimization for Sustainable Development, M. S. Manshahia et al., Eds. Hoboken, NJ: Scrivener Publishing LLC, 2022, pp. 271–350, Feb 2022.

[3] J. C. Caicedo, and S. Lazebnik, "Active object localization with deep reinforcement learning," In: Proc. IEEE Intl. Conf. Comput. Vision (ICCV), Santiago, Chile, 2015, pp. 1.

[4] T. Ilona, "Deep Reinforcement Learning in emotion recognition." Accessed: Oct. 30, 2023. [Online]. Available: https://github.com/ilonatommy/DLR_FacialEmotionRecognition

[5] G C. J. C. H. Watkins, and P. Dayan, "Q-Learning," Machine Learning. vol. 8, pp. 279–292, May 1992.

[6] N. Andalibi, and J. Buss, "The human in emotion recognition on social media: attitudes, outcomes, risks," In: Proc. Conf. Human Factors in Comput. Syst. (CHI), Honolulu, Hawaii, USA, 2020, pp. 1-16.

[7] L. Rhue, "Racial influence on automated perceptions of emotions," In: Proc. Soc. Sci. Res. Netw., Nov. 2018, pp. 1.

[8] C. Deng, X. Ji, C. Rainey, J. Zhang, and W. Lu, "iScience integrating machine learning with human knowledge," *Adaptive Behavior*, vol. 23, no. 11, 101656, Nov. 2020, pp. 1.

[9] J. Lin, Z. Ma, R. Gomez, K. Nakamura, B. He, and G. Li, "A review on interactive reinforcement learning from human social feedback," *IEEE Access*, vol. 8., pp. 120757–120765, July 2020.

[10] A. Maroti, "RBED: Reward Based Epsilon Decay," A. Maroti, "Rbed: Reward based epsilon decay," arXiv preprint arXiv:1910.13701, 2019, pp. 1-4.

[11] H. Van Hasselt, "Double Q-learning," In: Proc. 24th Conf. Neural Inform. Process. Syst. (NIPS), Vancouver, Canada, 2010, pp. 1-9.

[12] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," In: Proc. 30th Conf. Artif. Intell. (AAAI), Phoenix, Arizona, USA, 2016, pp. 1-13.

[13] J. Deng et al., "ImageNet: A Large-Scale Hierarchical Image Database." In: Proc. IEEE Conf. Comput. Vision Pattern Recognit. (CVPR), Miami, Florida, USA, 2009, pp. 248-255.

[14] C. Szegedy, V. Vanhoucke, S. Ioffe, and J. Shlens, "Rethinking the inception architecture for computer vision," In: Proc. IEEE Conf. Comput. Vision Pattern Recognit. (CVPR), Las Vegas, Nevada, USA, 2016, pp. 5-6.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," In: Proc. IEEE Conf. Comput. Vision Pattern Recognit. (CVPR), Las Vegas, Nevada, USA, 2016, pp. 3-4.

[16] M. Sandler et al., "MobileNetV2: inverted residuals and linear bottlenecks," In: Proc. IEEE Conf. Comput. Vision Pattern Recognit. (CVPR), Salt Lake City, Utah, USA, 2018, pp. 4-5.

[17] "FERG - V7 Open Datasets." Accessed: Oct. 30, 2023. [Online]. Available: https://www.v7labs.com/open-datasets/ferg