TAMER-ER: Augmenting the TAMER Framework with Expression Recognition

Benjamin Bienz
University of Colorado Boulder
Boulder, Colorado
benjamin.bienz@colorado.edu

Michael Lauria
University of Colorado Boulder
michael.lauria@colorado.edu

Vikas Nataraja
University of Colorado Boulder
viha4393@colorado.edu

Saumya Sinha
University of Colorado Boulder
saumya.sinha@colorado.edu

Christine T. Chang
University of Colorado Boulder
christine.chang@colorado.edu

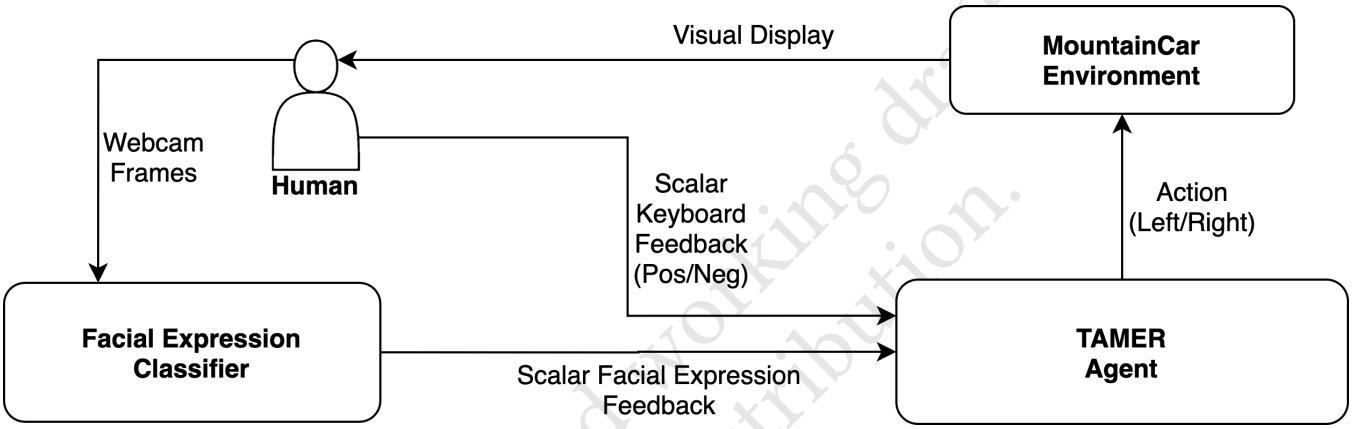

Figure 1: The TAMER-ER System.

ABSTRACT

In reinforcement learning, an agent interacts with an environment with the goal of maximizing a cumulative reward. However, this approach is limited by sampling inefficiencies and sparse or unknown reward signals. TAMER (Training an Agent Manually via Evaluative Reinforcement) [9] was introduced as a method for incorporating human feedback, which is provided as scalar value via keyboard input. The TAMER agent builds a model of the human reward function and chooses a greedy policy intended to maximize rewards. However, such explicit feedbacks can have a cognitive load on the human trainer. Through this work we want to investigate if it is possible to train a TAMER agent more naturally with social cues like facial expression. Facial expression recognition has been successfully utilized to provide feedback to robotic and autonomous systems. Thus, we propose TAMER-ER: TAMER with Expression Recognition where we augment the original TAMER framework with a classifier that recognizes human expressions. Our current work aims to build a facial expression recognition system and provide the outcomes to the TAMER agent as rewards. Future work includes fine-tuning this facial expression recognition system and providing the capability for including gestures and gaze, as well as conducting a human subjects study.

AFHRI '20, Jan – May, 2020, Boulder, CO

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

CCS CONCEPTS

• **Computing methodologies** → **Machine learning algorithms**; • **Computer systems organization** → **Robotic control**; • **Human-centered computing** → **Interactive systems and tools**.

KEYWORDS

TAMER, reinforcement learning, facial expression recognition, human-computer interaction

ACM Reference Format:
Benjamin Bienz, Michael Lauria, Vikas Nataraja, Saumya Sinha, and Christine T. Chang. 2020. TAMER-ER: Augmenting the TAMER Framework with Expression Recognition. In *Proceedings of AFHRI '20: Algorithmic Foundations of Human Robot Interaction (AFHRI '20).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Reinforcement learning (RL) is commonly used to train artificial intelligence agents to complete tasks in accordance with human preference. In reinforcement learning, an agent interacts with an

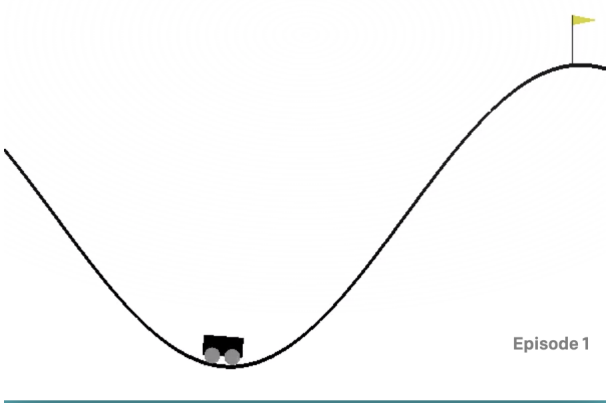

Figure 2: The MountainCar environment from OpenAI Gym.

environment with the goal of maximizing a cumulative reward. The agent acts according to a policy that is updated as it navigates the environment's state space and receives reward signals. While this approach can theoretically train in complex environments, in practice it is limited by extreme sample inefficiency (due to having to randomly explore the state space) and difficulties with environments that have a sparse or unknown reward signal. To mitigate both of these issues, a human teacher can be brought into the loop.

Knox and Stone [9] introduced the TAMER (Training an Agent Manually via Evaluative Reinforcement) framework as a method for algorithmically incorporating human feedback. In TAMER, human feedback is provided through scalar values, presented to the human test subject as "positive," "neutral," or "negative" signals via keyboard input. The TAMER agent builds a model of the human evaluation and runs a policy that aims to maximize the positive signal. Knox and Stone tested their original TAMER framework on Tetris and MountainCar; future iterations of this framework were tested on ATARI Bowling [22], Maze, Taxi, Car Robot [1], and robot navigation tasks in simulation [11].

While TAMER can successfully and efficiently train an agent, it relies on a system where a trained user can provide direct numerical feedback to a robot. Our experiment introduces TAMER-ER (TAMER with Expression Recognition), which seeks to expand the TAMER model by supplementing or replacing the keyboard-provided scalar feedback with additional feedback based on facial expression recognition. Specifically, we ask the following research questions:

RQ1: Can we improve the training of an autonomous agent via TAMER by augmenting the feedback signal with input from human facial expressions?

RQ2: Can we replace the keyboard-provided scalar feedback in TAMER with adjusted scalar values based on human facial expressions?

To investigate these questions, we implemented the original TAMER framework, trained a facial expression classifier for webcam video frames, and integrated these systems to train an agent that successfully plays the Open AI Gym version of MountainCar [18] (Fig. 2). A human subject (the "trainer") sits at a computer while watching the autonomous TAMER agent (the "agent") play MountainCar. The facial expression classifier takes snapshots of the trainer's expressions from the webcam. These images are classified as positive or negative, and this input (1 or -1, respectively) is then provided to the TAMER agent. The trainer also provides positive and negative feedback to the agent by using keyboard input as in the original TAMER experiment [9]. This keyboard input is mapped to the scalar values 1 and -1 utilized by the TAMER agent.

In the first iteration of our experiment where we investigate RQ1, we use the keyboard input from humans as a ground truth to predict a scalar reward from their facial expressions. This scalar reward augmented with the keyboard input trains the TAMER agent. We compare it with TAMER, where we only take the keyboard inputs. We separately train them from the same starting point, and the environmental rewards obtained at different training episodes are compared.

In the second iteration of our experiment, to investigate RQ2, we remove the keyboard input out of the system and train the TAMER agent using only the scalar reward given by the facial expressions.

Our hypotheses are as follows:

H1: Augmenting TAMER with facial expression classification will improve the agent's training, as we are effectively providing additional inputs for the model.

H2: With a basic implementation of facial expression classification alone, we will not be able to achieve the same learning rates as with TAMER. Because of the rapid learning rates achieved by TAMER, it would take a more finely tuned facial expression classifier to achieve results on par with TAMER.

2 RELATED WORK

The original TAMER framework was proposed and tested in 2009 [9]. As described in the introduction, TAMER trains on human feedback supplied by keyboard inputs representing "positive," "neutral," or "negative." The TAMER agent builds a model of the human reward signal and chooses a greedy policy that maximizes the reward. Many papers have followed that expand on this framework. We will discuss some of them here, as well as address other related research such as facial expression recognition.

VI-TAMER [10] uses value iteration to learn a value function with which it then selects actions. While the discount factor γ is set to 0 in TAMER, VI-TAMER sets γ close to 1, similar to traditional RL and is hence not myopic like TAMER.

In the implementation of Deep TAMER [22], the original TAMER framework is altered in a number of ways. First, Deep TAMER uses a deep neural network function approximation scheme. Second, it uses a different loss function than the one in the original TAMER; the new loss function "minimize[s] a weighted difference between the human reward and the predicted value for each state-action pair individually." Third, the model learns from each state-action pair multiple times rather than only once by utilizing a feedback replay buffer. Finally, Deep TAMER utilizes this feedback replay buffer only to address sparse feedback, and not to overcome instability during learning. The contribution of Deep TAMER is to adapt the TAMER framework on high-dimensional state space such as images.

DQN-TAMER [1] combines the sparse rewards of deep Q-learning with the frequent rewards of human feedback in TAMER to create

Action Unit	Description
2	Outer eyebrow raiser
4	Eyebrow lowerer
5	Upper eyelid raiser
6	Cheek raiser
12	Lip corner puller
14	Dimpler
17	Chin raiser
18	Lip puckerer
20	Lip stretcher
25	Lips part
26	Jaw drop

Table 1: Action Units (AUs) utilized in TAMER-ER as currently implemented. Adapted from [6].

one cohesive algorithm. DQN-TAMER also utilized solely human facial expressions to provide feedback to a "Car Robot," classifying facial expressions using a CNN. These expressions were mapped to positive, negative, and neutral, which the DQN-TAMER agent utilized to learn its reward function and reach the goal. However, DQN-TAMER uses MicroExpNet [24] for categorizing emotions into 8 categories: *neutral*, *anger*, *contempt*, *disgust*, *fear*, *happy*, *sadness*, and *surprise*. *Happy* was assigned a scalar value of 1, while all other categories except *neutral* were assigned values of -1. Alternately, facial action units (AUs) indicate individual and specific expressions on the face (see Table 1). Because people may not always provide full categorizable emotions, in TAMER-ER we instead use (AUs) with facial landmarks to map AUs to rewards.

Another instance of facial inputs being used with TAMER is [13], initially reported in an extended abstract [12]. However, this study focused on using an element of competition to incite more expressive faces, as well as investigating whether telling people to use facial expressions would increase their expressiveness. The expressions were not used in real time to inform the TAMER agent, rather they were later utilized to train a CNN-RNN model to predict human reward. This study further encouraged us to utilize AUs rather than categorized emotions for training, as in [13] the participants were significantly more expressive when either asked to compete with others performing the same task or when explicitly asked to be more expressive. In real world applications such as social robotics or even training an assistive robot for final assembly, a human trainer may not always be expressive to the extreme; we want to create a system that allows for relatively normal expressiveness. As Thomaz and Breazeal [20] elucidate regarding Socially Guided Machine Learning, "people will teach machines through a social and collaborative process and shall expect machines to engage in social forms of learning." We aim to show that it is possible to train a TAMER agent from online facial expression recognition, promoting a more social and natural form of learning.

Facial expression recognition has copious applications in human-robot and human-computer interaction, and many researchers have investigated classification of facial expressions for this purpose [3, 17, 21]. It is a common practice to track specified "facial landmarks," as in [8], [17], and [21], in order to determine the different positions of points on a person's face. The relative locations of these landmarks indicate, for example, whether an eyebrow has been raised or whether the lips have curled down. These positions indicate particular facial expressions. Broekens [4] demonstrates the positive effects of using human affect to augment a reinforcement learning agent, and use their results to advocate for the use of a human-in-the-loop system.

3 METHODS

TAMER-ER consists of multiple stages and components. First we implemented a version of the original TAMER algorithm that can be trained to play MountainCar (Section 3.1). Next we chose a dataset of human faces via webcam footage, as this most closely resembles our data to be collected. We trained a convolutional neural network (CNN) on this dataset (Section 3.3). Then we implemented our own facial expression detection for the human trainer, which utilizes the trainer's webcam. Frames are cropped and classified, and then scalar values of 1 for positive and −1 for negative are provided as training feedback to the TAMER agent (Section 3.2 and 3.3). Each of these are described in detail below, and their interactions are modeled in Figure 1.

Algorithm 1: TAMER Algorithm

Input: stepSize

$ReinfModel.init(stepSize)$;
$\overrightarrow{s} \leftarrow \overrightarrow{0}$;
$\overrightarrow{f} \leftarrow \overrightarrow{0}$;
while *true* **do**
  $h \leftarrow getHumanReinfSincePreviousTimeStep()$;
  if $h \neq 0$ **then**
    $error \leftarrow h - ReinfModel.predictReinf(\overrightarrow{f})$;
    $ReinfModel.update(\overrightarrow{f}, error)$;
  $\overrightarrow{s} \leftarrow getStateVec()$;
  $a \leftarrow$
   $argmax_a(ReinfModel.predict(getFeatures(\overrightarrow{s}, a)))$;
  $\overrightarrow{f} \leftarrow getFeatures(\overrightarrow{s}, a)$;
  $takeAction(a)$;
  wait for next timestep;

3.1 TAMER Implementation

First, a Q-learning [23] agent was implemented and tested on MountainCar [18]. An SGD Regressor with an RBF sampling preprocessing step [19] was chosen to approximate the $Q(s, a)$ function; this method has the benefits of incremental training and low computational cost while still allowing a nonlinear function to be learned. This agent was then converted to TAMER following the protocol referenced in the original TAMER paper [9], which consisted of the following changes:

- The environment render and the agent's current action is displayed explicitly via text on the screen (the environment render alone can be ambiguous).
- The user can input positive and negative rewards via the keyboard using the 'A' and 'W' keys respectively.

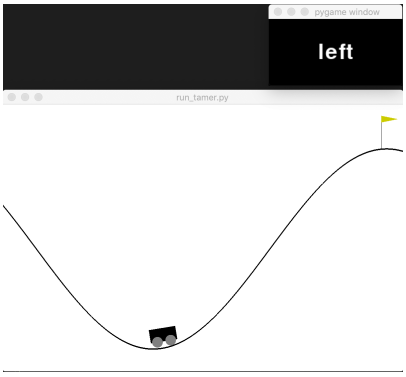

Figure 3: The MountainCar environment with agent's current action displayed.

- The environmental reward is replaced with user feedback (1 for positive, -1 for negative).
- The $H(s, a)$ (formerly $Q(s, a)$) function updates only for nonzero rewards.
- Environment timestep length was added as an adjustable parameter.

We found that by selecting a timestep long enough for the user to react to the current action we did not need to implement any sort of credit assignment function (which is a part of the original TAMER [9] implementation). We found empirically that a timestep of 0.2 - 0.5 seconds length was optimal for training (any longer and it is hard to intuit the agent's velocity from the render). With this configuration of TAMER it was possible to successfully train a MountainCar agent in less than one episode.

3.2 Facial Expression Recognition: Dataset

We explored a number of facial expression recognition data sets including Microsoft FER [2], Cohn-Kanade CK+ data set [14], and Affectiva's AM-FED+ data set [15]. The FER and the CK+ data sets are similarly structured in that they provide images mapped to emotions such as happy, sad, and angry. The Cohn-Kanade (CK) data set that preceded CK+ provided the first benchmark for automated facial action unit coding with a large number of posed expressions coded with facial action unit codes and recorded intensities. The Facial Action Coding System is a comprehensive, anatomically based system for describing all visually discernible facial movement. It breaks down facial expressions into individual components of muscle movement, called Action Units (AUs). Facial action units were introduced by Swedish anatomist Carl-Herman Hjortsjö [7], and later updated by Paul Ekman and Wallace Friesen [5]. The FER data set does not provide any data about facial action units and instead directly classifies images to one of 8 emotion categories - happiness, surprise, sadness, anger, disgust, fear, contempt and neutral. A significant drawback to using the FER data set for our proposed method is the extremity of the emotions in the data set. Most images are unrealistically extreme and unnatural which would not generalize to our experiment where we record a person viewing the MountainCar game and cannot expect such drastic emotional changes. Additionally, both these data sets provide images of very low resolution (48x48) which is challenging when adapting to webcam footage resolution which is much higher and necessary for our proposed method. Despite these challenges, we attempted cropping to the just the face on of our webcam frames, then reducing its dimensions to pass it through a pre-trained classifier (on the FER dataset). However, using standard libraries and off-the-shelf detectors for face detection made the system slow and laggy when training TAMER. It is also worth noting that emotions are combinations of

Ultimately, we used the AM-FED+ data set which is an extension of the AM-FED data set [16]. It has 545 videos of participants, and the majority of them are labeled with appropriate facial action units using FACS. To collect the data, the expressions and reactions of participants watching video advertisements were recorded and then manually labeled by FACS certified coders. Our final formatted data set has 15 classes of facial action units (AUs) (see Table 1), each corresponding to a different facial expression. Emotions can be extrapolated from AUs using specific combinations of these AUs.

3.3 Facial Expression Recognition: Training

Using facial action units as the ground truth, we first implemented a standard convolutional neural network composed of 6 convolutional layers and 3 dense layers. As the input, we fed the raw images from the formatted AM-FED+ data set and trained the model with binary cross entropy (BCE) loss since there can be multiple AUs present in a single image; we wanted the model to learn features independently given that this is a multi-label classification problem. However, the model could not overcome the different backgrounds in sets of images from the video stream, and even with cropped images it could not learn the appropriate features required to map faces and facial expressions to AUs. As a result, we switched to using facial landmark features as a way to reduce background noise.

To extract facial landmarks from images, we used an off-the-shelf facial detection and prediction system that produces 68 features as facial landmarks that are focused on eyebrows, eyes, mouth and chin. After running the detector, our data set was reduced to around 3,100 images and the AU distribution of the set is shown in Figure 4. Since the images are 320x240, we also resized the frames during evaluation to the same size to scale the landmark coordinates.

To learn the relationship between facial landmarks and facial action units, we used a 3 layer dense network with ReLU activation for the first 2 layers and a sigmoid function activation for the final layer. The sigmoid function produces a probability score between 0 and 1 for each of the AUs.

Once we have our AUs predicted with some accuracy, we want to map it to scalar feedback signals to provide rewards for training TAMER. To begin with, we used a simple heuristic function that maps AU probabilities to emotions, these emotions are then mapped to scalar values. For example, if *cheek raiser* AU06 and *lip corner puller* AU12 have the high probabilities from the classifier output, the heuristic takes that to mean "happiness" which then gets translated to a scalar feedback of +1. Since there are three possible feedback signals (neutral, positive, and negative), we can treat this mapping as a classification problem. We intend to implement traditional machine learning classifiers such as logistic regression and random forest. The value of these methods is their simplicity, explainability, and the option to estimate feature importance (here,

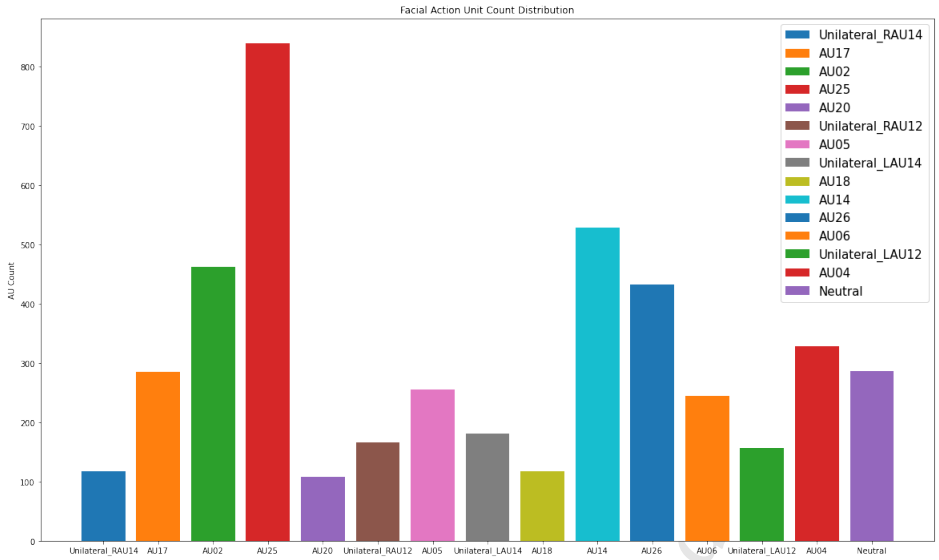

Figure 4: The distribution of facial action units in the final formatted data set.

the importance of a given AU). In addition, we plan to use at an online algorithm to allow us to improve the classification over the course of a single training session, and for a particular user.

4 EVALUATION

In the human subjects evaluation that has not yet been implemented, an individual participant will have a computer with a monitor on which to watch the game and a keyboard to provide positive and negative feedback. They will also be video recorded. They will be told that the more accurate the feedback they can provide to the computer that is playing the game, the more points they will accumulate. Their goal is to get as many points as possible. In our control condition, only the keyboard input will be used. In our experimental conditions, we will (1) utilize both the keyboard and expression recognition or (2) utilize only expression recognition to inform the TAMER agent. Using the video, our model will classify the facial expressions of the human, assign weights to key expressions, and provide the predicted scalar output as a reward to our framework. At the end of the human-facilitated training session, we have a learned reward function for our MountainCar task. We can obtain environmental rewards at different episodes of the training to see if we are learning a good policy and compare it against the standard TAMER implementation.

During evaluative testing, the authors have observed that as currently implemented, TAMER-ER does not show improved or comparable results to TAMER. Current observations and next steps are discussed in the following section.

5 CONCLUSION AND FUTURE WORK

In conclusion, as currently implemented, TAMER-ER does not show improved or comparable results to TAMER. Thus, neither of our hypotheses **H1** or **H2** can yet be shown as correct. However, the authors feel strongly that improvement can be made, and we intend to continue working on various aspects of the design as outlined here. One observation is that because original TAMER allows the agent to be trained so quickly, adding facial expression recognition to the keyboard input may not produce any measurable improvement. The current facial recognition model outputs are unable to produce good scalar feedback (+1 or -1) to train our TAMER agent. With a better FER model, though, it may be possible to replace or augment keyboard input with facial expressions.

We suspect not having enough training examples for each AU label can be potential reason for the current performance of the face classifier model. The Affectiva AM-FED+ data set while originally having 8,000 frames (or 545 video files) had to be reduced to 3,000 frames due to insufficient data for certain action units and missing data for some frames. This resulted in the final data set having significantly less *influential* action units. For example, AU05 corresponding to "Upper eyelid raiser" does not actually influence the final reward but had a significant number of frames in the data set.

One obvious approach to improving our facial expression training would be fine-tuning the model on a different data set, ideally captured from a webcam with participant reactions recorded and labeled with facial action units. This would improve the model's performance during real-time and offer more data points for evaluation. Another possible route we would like to explore is the use of LSTMs or Long Short Term Memory networks in combination with CNNs to capture temporal information which would enable the model to learn the changes with facial expressions with time. During the course of the entire run or even during an episode of MountainCar, tracking expressions over time might be beneficial to learn the importance of certain expression-action correspondences in the game thereby helping the agent learn a better policy.

We would also like to investigate if facial expression recognition can facilitate a shared autonomous system where a human has inadequate control and is assisted by an autonomous agent. We believe that humans will have a higher motivation to provide feedback in

this kind of environment, thus making them more expressive. This increase in facial expressions could provide a more complete and informative model for the TAMER agent, which would improve training rates over TAMER.

The next step for this project is to obtain Institutional Review Board approval and conduct this study on human subjects who are not the authors. We would like to obtain a large enough sample that we are able to draw some larger conclusions with respect to utilizing human facial expressions to train a TAMER agent. Lastly, we plan to incorporate more social cues like user's gestures or gaze to provide additional signals over the keyboard input and facial expressions for efficiently training TAMER.

6 ONLINE RESOURCES

All code for TAMER-ER can be found on our public repository: https://github.com/saumyasinha/learning-via-human-feedback

ACKNOWLEDGMENTS

Thank you to Dr. Brad Hayes for his valuable assistance and guidance.

REFERENCES

[1] Riku Arakawa, Sosuke Kobayashi, Yuya Unno, Yuta Tsuboi, and Shin ichi Maeda. 2018. DQN-TAMER: Human-in-the-Loop Reinforcement Learning with Intractable Feedback. *ArXiv* abs/1810.11748 (2018).

[2] Emad Barsoum, Cha Zhang, Cristian Canton-Ferrer, and Zhengyou Zhang. 2016. Training Deep Networks for Facial Expression Recognition with Crowd-Sourced Label Distribution. *CoRR* abs/1608.01041 (2016). arXiv:1608.01041 http://arxiv.org/abs/1608.01041

[3] M.S. Bartlett, G.C. Littlewort, I.R. Fasel, J. Chenu, T. Kanda, H. Ishiguro, and J.R. Movellan. 2004. Towards Social Robots: Automatic Evaluation of Human-Robot Interaction by Facial Expression Classification. In *Advances in Neural Information Processing Systems 16*, S. Thrun, L. K. Saul, and B. Schölkopf (Eds.). MIT Press, 1563–1570.

[4] Joost Broekens. 2007. Emotion and Reinforcement: Affective Facial Expressions Facilitate Robot Learning. In *Artifical Intelligence for Human Computing (Lecture Notes in Computer Science)*, Thomas S. Huang, Anton Nijholt, Maja Pantic, and Alex Pentland (Eds.). Springer, Berlin, Heidelberg, 113–132. https://doi.org/10.1007/978-3-540-72348-6_6

[5] Paul Ekman and Wallace V Friesen. 1978. *Facial action coding system: Investigator's guide.* Consulting Psychologists Press.

[6] Bryn Farnsworth. [n.d.]. Facial Action Coding System (FACS) - A Visual Guidebook. https://imotions.com/blog/facial-action-coding-system/ Library Catalog: imotions.com.

[7] Carl-Herman Hjortsjö. 1969. *Man's face and mimic language.* Studen litteratur.

[8] Vahid Kazemi and Josephine Sullivan. 2014. One millisecond face alignment with an ensemble of regression trees. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 1867–1874. https://doi.org/10.1109/CVPR.2014.241 ISSN: 1063-6919.

[9] W. Bradley Knox and Peter Stone. 2009. Interactively shaping agents via human reinforcement: the TAMER framework. In *Proceedings of the Fifth International Conference on Knowledge Capture (K-CAP '09)*. Association for Computing Machinery, Redondo Beach, California, USA, 9–16. https://doi.org/10.1145/1597735.1597738

[10] W. Bradley Knox and Peter Stone. 2015. Framing reinforcement learning from human reward: Reward positivity, temporal discounting, episodicity, and performance. *Artificial Intelligence* 225 (Aug. 2015), 24–50. https://doi.org/10.1016/j.artint.2015.03.009

[11] W. Bradley Knox, Peter Stone, and Cynthia Breazeal. 2013. Teaching agents with human feedback: a demonstration of the TAMER framework. In *Proceedings of the companion publication of the 2013 international conference on Intelligent user interfaces companion (IUI '13 Companion)*. Association for Computing Machinery, Santa Monica, California, USA, 65–66. https://doi.org/10.1145/2451176.2451201

[12] Guangliang Li, Hamdi Dibeklioglu, Shimon Whiteson, and Hayley Hung. 2016. Towards Learning from Implicit Human Reward: (Extended Abstract). In *Proceedings of the 2016 International Conference on Autonomous Agents 'I&' Multiagent Systems* (Singapore, Singapore) *(AAMAS '16)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1353–1354.

[13] Guangliang Li, Hamdi Dibeklioğlu, Shimon Whiteson, and Hayley Hung. 2020. Facial Feedback for Reinforcement Learning: A Case Study and Offline Analysis Using the TAMER Framework. arXiv:cs.HC/2001.08703

[14] Patrick Lucey, Jeffrey Cohn, Takeo Kanade, Jason Saragih, Zara Ambadar, and Iain Matthews. 2010. The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, CVPRW 2010*, 94 – 101. https://doi.org/10.1109/CVPRW.2010.5543262

[15] D. McDuff, M. Amr, and R. e. Kaliouby. 2019. AM-FED+: An Extended Dataset of Naturalistic Facial Expressions Collected in Everyday Settings. *IEEE Transactions on Affective Computing* 10, 1 (2019), 7–17.

[16] D. McDuff, R. el Kaliouby, T. Senechal, M. Amr, J. F. Cohn, and R. Picard. 2013. Affectiva-MIT Facial Expression Dataset (AM-FED): Naturalistic and Spontaneous Facial Expressions Collected "In-the-Wild". In *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 881–888.

[17] Philipp Michel and Rana El Kaliouby. 2003. Real time facial expression recognition in video using support vector machines. In *Proceedings of the 5th international conference on Multimodal interfaces (ICMI '03)*. Association for Computing Machinery, Vancouver, British Columbia, Canada, 258–264. https://doi.org/10.1145/958432.958479

[18] OpenAI. [n.d.]. Gym: A toolkit for developing and comparing reinforcement learning algorithms. https://gym.openai.com Library Catalog: gym.openai.com.

[19] Ali Rahimi and Benjamin Recht. 2007. Random features for large-scale kernel machines. In *Proceedings of the 20th International Conference on Neural Information Processing Systems (NIPS'07)*. Curran Associates Inc., Vancouver, British Columbia, Canada, 1177–1184.

[20] Andrea Lockerd Thomaz and Cynthia Breazeal. 2005. Socially Guided Machine Learning: Designing an Algorithm to Learn from Real-Time Human Interaction. (Dec. 2005), 10.

[21] Vivek Veeriah, Patrick M. Pilarski, and Richard S. Sutton. 2016. Face valuing: Training user interfaces with facial expressions and reinforcement learning. *arXiv:1606.02807 [cs]* (June 2016). http://arxiv.org/abs/1606.02807 arXiv: 1606.02807.

[22] Garrett Warnell, Nicholas Waytowich, Vernon Lawhern, and Peter Stone. 2018. Deep TAMER: Interactive Agent Shaping in High-Dimensional State Spaces. In *Thirty-Second AAAI Conference on Artificial Intelligence*. AAAI. https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16200

[23] Christopher J. C. H. Watkins and Peter Dayan. 1992. Q-learning. *Mach Learn* 8, 3 (May 1992), 279–292. https://doi.org/10.1007/BF00992698

[24] İlke Çuğu, Eren Şener, and Emre Akbaş. 2019. MicroExpNet: An Extremely Small and Fast Model For Expression Recognition From Face Images. *arXiv:1711.07011 [cs]* (Dec. 2019). http://arxiv.org/abs/1711.07011 arXiv: 1711.07011.