Hand Gesture Recognition using Deep Reinforcement Learning for In-vehicle Environments

Arman Hafizi arman.hafizi@uwaterloo.ca
Cheriton School of Computer Science - University of Waterloo - Waterloo, ON, CA

Editor: Arman Hafizi

Keywords: Hand Gesture Recognition, Deep Reinforcement Learning, Midair Gestures

Abstract

Hand Gesture Recognition (HGR) has been widely discussed in different fields. However, for in-vehicle interaction between the driver and infotainment system where accuracy, safety, and ease are priorities, some proposed systems lack such characteristics. Existing methods are base on either mathematical knowledge or machine learning which both have their limitations. We propose a Deep Reinforcement Learning (DRL) approach to train a classification system. Results indicate that such system can reach 94% accuracy in an acceptable time.

1. Related Work

The automotive industry is progressively moving to centralize the display and control of vehicle systems on a centre console screen (Reuters ·). Originally for "infotainment" like music and navigation, these are now frequently used for features like cabin climate control and vehicle status, and increasingly, even for critical functions like windshield wiper speed (Rapier). Likely due to familiarity and the manufacturing economy, a touchscreen is commonly used for primary input, but this requires the driver to momentarily redirect their visual focus away from the roadway (Ng et al., 2017). Speech control (Angelini et al., 2016; Cui et al., 2021a) and steering wheel buttons (Hassan et al., 2022; Philip Hwang et al., 2020) are potential eyes-free alternatives, but they have limitations. For instance, speech can be slow and unreliable, especially in loud environments, and it can disturb passengers (Jung et al., 2020; Cui et al., 2021b). Steering wheel buttons increase manufacturing complexity and cost, and they are typically reserved for dedicated vehicle functions like cruise control.

Gestures performed on the touchscreen or in midair can in theory be performed eyes-free, and midair gestures are already used in some high-end vehicles (Burns, 2019). BMW systems use an index-to-thumb pinch to delimit midair gesture motion from other hand movements, and has demonstrated some form of directional gesture (BMW, 2015). However, performing midair gestures for in-vehicle are more error prone and less accurate than touch gestures (Henderson et al., 2019). For touch interfaces, the user is aware of the cursor position whereas in midair gestures systems, it is usually unknown.

Current implementations of HGR are mostly based on either mathematical knowledge or machine learning. Simple theoretical knowledge such as distance and direction of the performed gesture are used for pattern matching (Zhao and Balakrishnan, 2004; Zhao et al., 2006). As it sounds logical to have such assumptions, it is likely that other insights about

this performance are ignored. To give an example, recognizing pinch gesture only based on the markers distance is basic as some users may do the pinch after they have already dragged a little directional gesture (Ren and O'Neill, 2012). Others might start a bit lower as they see the prompted tasks are mostly upwards to save themselves more space. Moreover, applying machine learning approaches are time-consuming and usually needs a noticeable number of labeled data. Furthermore, these trained models are all fixed and cannot be further trained since there is no feedback provided for in-vehicle environments.

Several works have previously investigated pattern recognition using RL methods. Goyal et al. (2015) applied machine learning methods to train a system for HGR based on static webcam data. Once all machine learning models were trained, they used those parameters as initial parameters to further improve those models. Not only did they use static gestures, but they chose the best 5 or 10 gestures to report that resulted in the highest accuracy. Seok et al. (2018) more directly implemented a DRL based classifier for HGR. They used EMG signals of three different dynamic gestures: drawing a circle, square, and triangle. The results indicated a higher performance of the method with Convolutional Neural Network (CNN) core than Long Short-Term Memory (LSTM) core. More recently, Vásconez et al. (2021) applied a Q-learning approach to classify hand gestures. They used windowed EMG signals to classify five dynamic gestures and showed that the model could reach a reasonable accuracy. However, they only trained a binary classification model where it was able to predict whether or not a gesture is performed.

We implement a DRL approach, by gathering information from many samples to build a model that can overcome traditional methods limitations. We aim to train a multi-classification model based on DRL that is quickly trained, accurate, and can be further tuned with online data.

2. Method

2.1 Data Acquisition

Midair gestures are recognized mostly using ray casting. In our experiment, we used Vicon Tracker [1] system for midair gesture recognition. The system operates by creating a 3D model of environment using data gathered from attached markers. We attached three marker on thumb, index, and middle fingers to track hand movements.

Ten midair hand gestures that are the most favorite for in-vehicle interaction were chosen for this study. Simple whole hand swipes to right, left, up, down, front, and back were considered as directional gestures. For rolling gestures, drawing a circle clockwise and counter clockwise were chosen. To include static gestures, we decided to use "Ok" and "Stop" postures (Figure 1).

For each gesture, forty samples were recorded, i.e. 400 samples in dataset. The gestures did not share the same completion time and each had variable number of snapshots of Vicon data with a mean of 475 (ranging from 352 to 521). However, each snapshot of Vicon data consisted of coordinates of markers in space, i.e. 3 markers $\times$ 3 axis. Thus each sample x_{raw} from our dataset X_{raw} had a shape of $(n, 3, 3)$ and corresponding y_{raw} label in range $[0, 9]$.

1. `www.vicon.com/software/tracker`

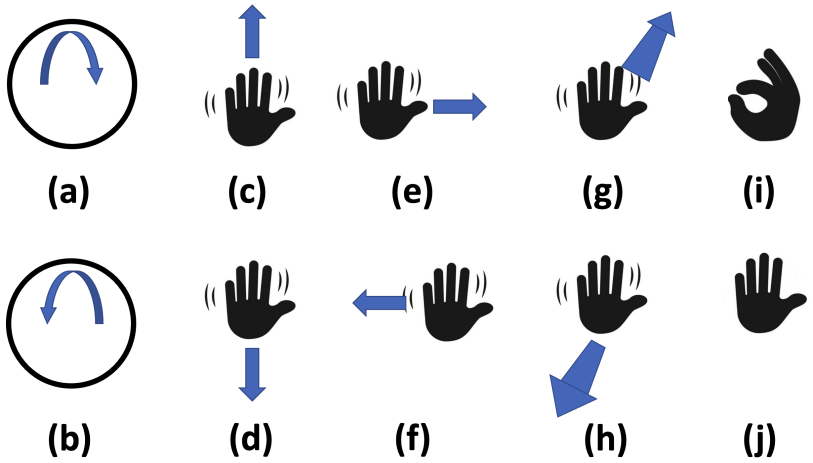

Figure 1: Hand Gesture Set: (a) clockwise circle; (b) counter clockwise circle; (c) swipe up; (d) swipe down; (e) swipe right; (f) swipe left; (g) swipe forward; (h) swipe backward; (i) "Ok" posture; (j) "Stop" postures.

2.2 Pre-processing

First, we normalized our data using standard deviation of each sample:

$$x = \frac{x_{raw} - \mu(x_{raw})}{\sigma(x_{raw})} \tag{1}$$

That is necessary to remove the exact position of the markers and only consider the relative movement of the hand. Moreover, to feed the gradual behaviour of data to our model, we split each sample to sliding windows with fixed strides that contain a specific number of snapshots. The same label of sample was assigned to all windows in that sample. We evaluated different window sizes W, and stride S in our study.

2.3 Deep Q-Network Classification

Q-learning is an RL algorithm that does not know how the policy experience was generated. Since we do not have access to environments states, we assume that only observations are given. Therefore, we use a Partially Observable Markov Decision Process (POMDP) instead of a simple MDP. Q-learning method based on action A and observation O is mentioned below:

$$Q_{new}(O_{t-1}, A_{t-1}) \leftarrow Q_{old}(O_{t-1}, A_{t-1}) + \alpha(R_t + \gamma \cdot max_a[Q(O_t, a)] - Q_{old}(O_{t-1}, A_{t-1})) \tag{2}$$

where $Q(O_t, A_t)$ is the Q-values, and α and γ are learning rate and discount factor respectively.

More specifically, we implement Q-learning via a Deep Neural Network (DNN) to facilitate handling large data. Approximating the Q-function by output gradient and updating weights using:

$$\omega \leftarrow \omega + \alpha(R_t + \gamma \cdot max_a[Q(\omega, O_t, a)] - Q_{old}(\omega, O_{t-1}, A_{t-1})) \cdot \nabla Q(\omega, O_{t-1}, A_{t-1}) \tag{3}$$

Having a look at interaction of the agent and environment (Figure 2), we are able to define our main variables.

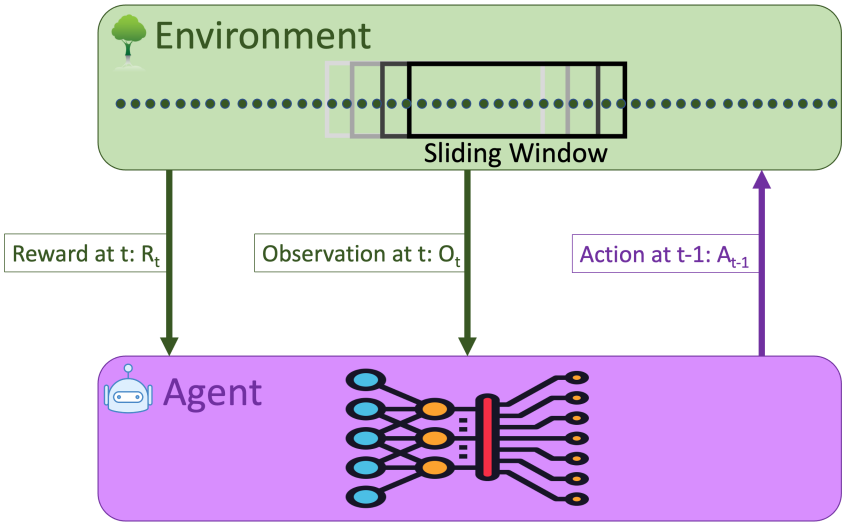

Figure 2: General Architecture of the Model

Agent is the element that decides which actions to choose that maximizes the total reward. Agent ideally wants to predict all labels of observations in one sequence correctly.

An **Observation** O_t in our environment is the window that agent is faced at a specific time. This gives some information about the actual hidden state to the agent. As the last observation is passed to the agent, the episode ends.

An **Action** A_t is the current classification that agent makes using O_t. Since we had 10 gestures in our dataset, the action space is an integer value between 0 and 9 corresponding to clockwise circle, counter clockwise circle, swipe right, swipe left, swipe up, swipe down, swipe forward, swipe backward, "OK" posture, and "Stop" posture

Reward is defined in case of correct or incorrect prediction of an action by agent. If the chosen action A_t by agent is the correct label of the observation O_t, it receives a $R = +1$ and in case of incorrect prediction, a $R = -1$ is given.

3. Evaluation

For our Q network architecture, we used a deep sequential model with 3 linear layers with 500 hidden nodes and ReLU functions. Adam optimizer with learning rate $lr = .0005$ was chosen for the optimization part. A floating epsilon updating was also used for ϵ-greedy policy function starting at 1 and ending at .01. We ran models with window size $W \in \{50, 75\}$ and stride $S \in \{10, 20\}$ for 10,000 episodes that updated every 10 episodes. Discount factor γ was always set to .99. To reduce stochasticity, average curves of 5 runs corresponding to 5 random seeds are reported.

We calculated the average rewards of the last 25 episodes for each case. Figure 3 indicates that curves for larger strides ($S = 20$) showed a more robust behaviour than smaller strides ($S = 10$). Not only $S = 20$ curves had less variance, but it almost always

continued to gradually improve. On the other hand, although $S = 10$ curves rose in the first 250 episodes, but they suddenly decreased and showed an unstable behaviour. Note that changing the window size from $W = 50$ to $W = 75$ did not affect the results for stride $S = 20$, but increased the fluctuations for $S = 10$.

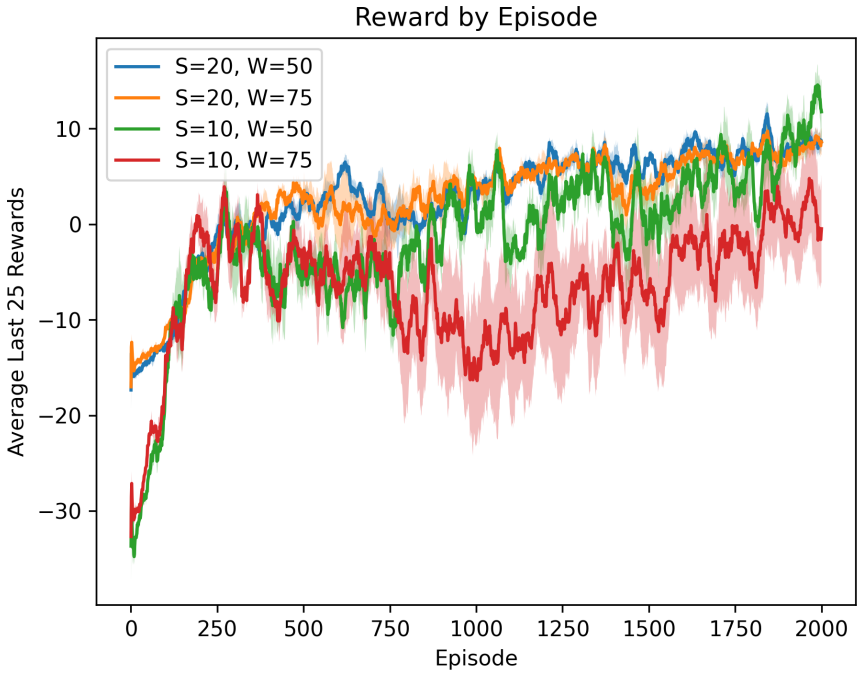

Figure 3: Average Rewards of the Last 25 Episodes by Episode

To evaluate the models' accuracy, we compared the actual gesture label to the most common predicted label (action) in an episode. Note that if model had failed to learn the environment behaviour, it would have returned an average accuracy of 10% since we have 10 classes in our dataset. However, the model improved the accuracy by far. Figure 4 shows that curves for larger strides ($S = 20$) were more accurate and robust than smaller strides ($S = 10$). Despite initial good accuracy of $S = 10$ curves, they failed to keep the same improvement rate. On the other hand, $S = 20$ curves continued to improve to reach 94% accuracy.

4. Conclusion

We designed a Deep Reinforcement Learning method of Hand Gesture Recognition problem for in-vehicle systems. Using a deep classifier based on Q-learning, we showed that the agent can learn the environment to get high acceptable rewards and predict the correct label with 94% accuracy. The results showed that sliding on the data with a larger stride on divided windows train a more robust agent to predict the classes. Future research can

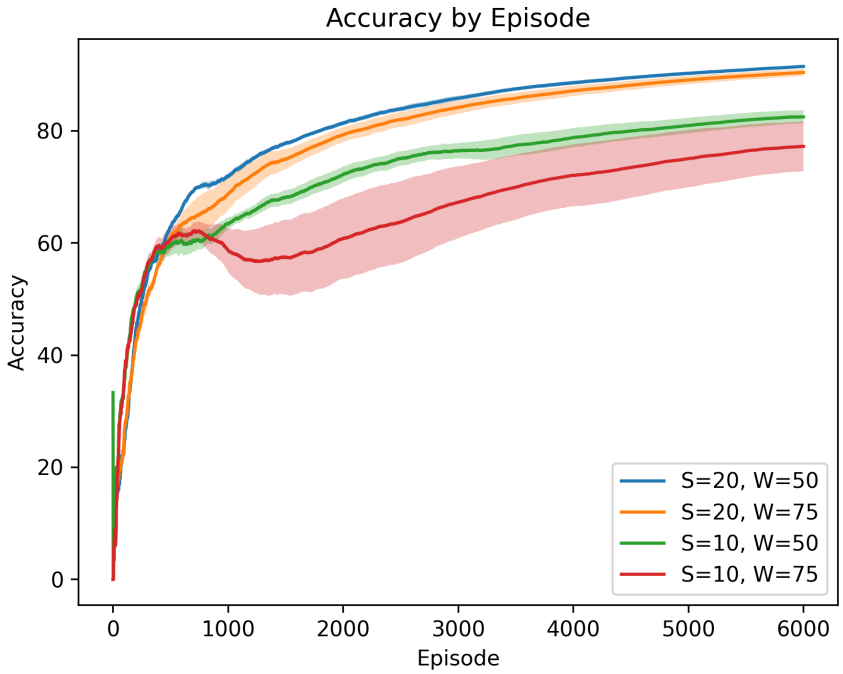

Figure 4: Classification Accuracy by Episode

be on optimizing the deep core of the method and assign parameters based on the specific environment.

References

Leonardo Angelini, Jürgen Baumgartner, Francesco Carrino, Stefano Carrino, Maurizio Caon, Omar Khaled, Jürgen Sauer, Denis Lalanne, Elena Mugellini, and Andreas Sonderegger. Comparing gesture, speech and touch interaction modalities for in-vehicle infotainment systems. In *Actes de la 28ième conférence francophone sur l'Interaction Homme-Machine*, pages 188–196, 2016.

BMW. Pinch and drag gesture — bmw genius how-to, 2015. URL `https://www.youtube.com/watch?v=jzqoTu_q78c`.

Matt Burns. BMW's magical gesture control finally makes sense as touchscreens take over cars, November 2019. URL `https://techcrunch.com/2019/11/04/bmws-magical-gesture-control-finally-makes-sense-as-touchscreens-take-over-cars/`.

Zhitong Cui, Hebo Gong, Yanan Wang, Chengyi Shen, Wenyin Zou, and Shijian Luo. Enhancing interactions for in-car voice user interface with gestural input on the steering wheel. In *13th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, AutomotiveUI '21, page 59–68, New York, NY, USA, 2021a.

Association for Computing Machinery. ISBN 9781450380638. doi: 10.1145/3409118.3475126. URL https://doi.org/10.1145/3409118.3475126.

Zhitong Cui, Hebo Gong, Yanan Wang, Chengyi Shen, Wenyin Zou, and Shijian Luo. Enhancing interactions for in-car voice user interface with gestural input on the steering wheel. In *13th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, pages 59–68, 2021b.

Arpit Goyal, Andrew Ciambrone, and Hossameldin Shahin. Hand gesture recognition with batch and reinforcement learning. 2015.

Waseem Hassan, Ahsan Raza, Muhammad Abdullah, Mohammad Shadman Hashem, and Seokhee Jeon. Hapwheel: Bringing in-car controls to driver's fingertips by embedding ubiquitous haptic displays into a steering wheel. *IEEE Transactions on Intelligent Transportation Systems*, 2022.

Jay Henderson, Sachi Mizobuchi, Wei Li, and Edward Lank. Exploring cross-modal training via touch to learn a mid-air marking menu gesture set. In *Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services*, pages 1–9, 2019.

Jingun Jung, Sangyoon Lee, Jiwoo Hong, Eunhye Youn, and Geehyuk Lee. Voice+ tactile: Augmenting in-vehicle voice user interface with tactile touchpad interaction. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2020.

Alexander Ng, Stephen A Brewster, Frank Beruscha, and Wolfgang Krautter. An evaluation of input controls for in-car interactions. In *Proceedings of the 2017 CHI conference on human factors in computing systems*, pages 2845–2852, 2017.

TK Philip Hwang, Yao-Tin Huang, and Pin-Chieh Kuo. Employing natural finger positioning strategy for improving blind-positioning of steering wheel mounted switches. In *International Conference on Applied Human Factors and Ergonomics*, pages 85–91. Springer, 2020.

Graham Rapier. German court ruled Tesla's touchscreen windshield wiper controls are the same as texting while driving. URL https://www.businessinsider.com/tesla-touchscreen-wiper-controls-akin-texting-driving-german-court-2020-8.

Gang Ren and Eamonn O'Neill. 3d marking menu selection with freehand gestures. In *2012 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 61–68. IEEE, 2012.

Thomson Reuters ·. Biggest trend in new car technology? super-sized screens | CBC news. URL https://www.cbc.ca/news/science/ces-trend-48-inch-screen-car-1.4972574.

W Seok, Youngjoo Kim, and Cheolsoo Park. Pattern recognition of human arm movement using deep reinforcement learning. In *2018 International Conference on Information Networking (ICOIN)*, pages 917–919. IEEE, 2018.

Juan Pablo Vásconez, Lorena Isabel Barona López, Ángel Leonardo Valdivieso Caraguay, Patricio J Cruz, Robin Álvarez, and Marco E Benalcázar. A hand gesture recognition system using emg and reinforcement learning: A q-learning approach. In *International Conference on Artificial Neural Networks*, pages 580–591. Springer, 2021.

Shengdong Zhao and Ravin Balakrishnan. Simple vs. compound mark hierarchical marking menus. In *Proceedings of the 17th annual ACM symposium on User interface software and technology*, pages 33–42, 2004.

Shengdong Zhao, Maneesh Agrawala, and Ken Hinckley. Zone and polygon menus: using relative position to increase the breadth of multi-stroke marking menus. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 1077–1086, 2006.