



HAL
open science

Sketch Recognition for Interactive Game Experiences Using Neural Networks

Elif Hilal Korkut, Elif Surer

► **To cite this version:**

Elif Hilal Korkut, Elif Surer. Sketch Recognition for Interactive Game Experiences Using Neural Networks. 20th International Conference on Entertainment Computing (ICEC), Nov 2021, Coimbra, Portugal. pp.393-401, 10.1007/978-3-030-89394-1_31 . hal-04144389

HAL Id: hal-04144389

<https://inria.hal.science/hal-04144389v1>

Submitted on 28 Jun 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



This document is the original author manuscript of a paper submitted to an IFIP conference proceedings or other IFIP publication by Springer Nature. As such, there may be some differences in the official published version of the paper. Such differences, if any, are usually due to reformatting during preparation for publication or minor corrections made by the author(s) during final proofreading of the publication manuscript.

Sketch Recognition for Interactive Game Experiences Using Neural Networks

Elif Hilal Korkut¹[0000-0002-8836-6685] and Elif Surer¹[0000-0002-0738-6669]

Department of Modeling and Simulation, Graduate School of Informatics,
Middle East Technical University, 06800 Ankara, Turkey
`elif.korkut@metu.edu.tr`, `elifs@metu.edu.tr`

Abstract. Human freehand sketches can provide various scenarios to the interfaces with their intuitive, illustrative, and abstract nature. Although freehand sketches have been powerful tools for communication and have been studied in different contexts, their capacity to create compelling interactions in games is still under-explored. In this study, we present a new game based on sketch recognition. Specifically, we train various neural networks (Recurrent Neural Networks and Convolutional Neural Networks) and use different classification algorithms (Support Vector Machines and k-Nearest Neighbors) on sketches to create an interactive game interface where the player can contribute to the game by drawing. To measure usability, technology acceptance, immersion, and playfulness aspects, 18 participants played the game and answered the questionnaires composed of four different scales. Technical results and user tests demonstrate the capability and potential of sketch integration as a communication tool to construct an effective and responsive visual medium for novel interactive game experiences.

Keywords: sketch recognition · neural networks · deep learning · interactive interfaces · game experiences.

1 Introduction

As an abstract representation of human expression, sketches have been used in various fields, from art to engineering, throughout history. A sketch can convey information or high-level ideas with minimum detail. Due to success in deep learning techniques within the Human-Computer Interaction (HCI) community, sketching has become an emerging research topic. Although hand-drawn sketches show promising results, especially for educational purposes, their potential for games is waiting for exploration.

Human freehand sketches have been studied in various domains. For example, a comparative study conducted by Melanie et al. [1] showed that Recurrent Neural Networks (RNNs) performed better than CNNs. Sketch-based interaction is an old concept for games that involve drawing, such as Pictionary and Hangman.

Later, sketch-based interaction has also been explored by digital games. To encourage and motivate people, ZenSketch [12] was developed to improve freehand sketching linework, which translated features of sketches into game mechanics.

This study proposes new game mechanics powered by neural networks to create a responsive, generative, and interactive experience. Different types of neural networks are trained on selected classes of the Quick Draw dataset [6] and deployed into the Unity 3D game engine. A 3D maze game is generated, which uses sketched objects drawn by the player in the game world. Furthermore, we demonstrate the capability of sketch recognition in the field of HCI. To measure usability, technology acceptance, immersion, and playfulness aspects, 18 participants played the game and answered the questionnaires on system usability, gamer profile, technology acceptance, and gameful experiences. Our technical contribution lies in testing and creating various models and methods to enable seamless interaction while creating a novel and responsive game experience. This study aims to facilitate game ideas that create a visual dialogue between the game, game mechanics, and players where players can actively develop the narrative itself by sketching.

2 Methods

2.1 Game Design

A 3D maze game, Sketchscape, is implemented using the Unity game engine to test and validate the proposed interaction method. The game starts in a room and gives hints to the player to solve the puzzles and overcome challenges to reach the exit. There are three types of object groups; non-interacting objects, duty objects, and spawned objects. Non-interacting objects (key, eyeglasses, compass) directly affect the game world without appearing in the world or appear only to guide players without physical interaction. Duty objects (boat, bridge, air balloon) are only instantiated in certain positions when they are drawn and submitted. For the spawned objects, classes were defined as the circle, triangle, square, and due to their primitiveness, they provide easy and fast interaction. Spawned objects (cube, triangular prism, sphere) are instantiated in front of the player whenever they are drawn. Players can use spawned objects to destroy enemies, pass through the water, or use them as a ramp. The primary aim of those objects is to give the player freedom to act spontaneously. Throughout the game, there are hints for each object. To disentangle the current situation, the player needs to draw the correct object. For example, if the player needs to open a locked door and the sketch of a key is submitted, the door will open automatically (Fig. 1). User interactions should be intuitive and give necessary feedback to the player. Therefore, we avoided complicated user interactions and used standard character controls while creating a simple interface that players can draw with the cursor and get feedback from the game as soon as they submit their sketch. A cursor-controlled drawing feature is added, and the canvas is arranged according to the size of the sketches. The interface includes two canvas options, full-screen mode and side panel. The full-screen drawing panel opens

via the right mouse button, and when the player hits submit button, it is closed automatically. Hints are given to the player in descriptive text format.

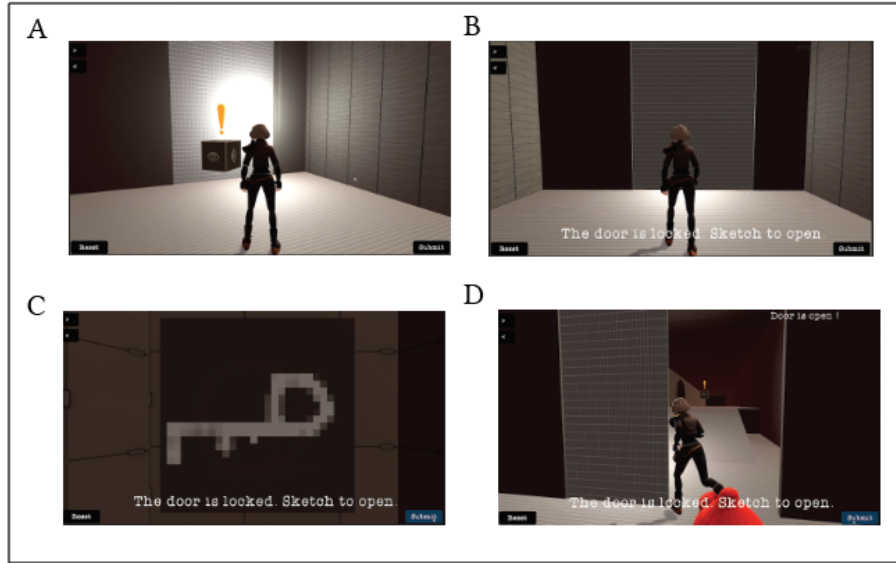


Fig. 1. Example Gameplay A. In a room with a hint B. Taking the hint, and text appears on the screen. C. Drawing a key on canvas. D. The door opens.

2.2 Data Exploration

Quick Draw dataset [6], consisting of 345 different classes with over 50 million sketches, includes more than 100000 items for each class. It includes positional information of points, temporal order, and grouping of points as strokes. For this project, only nine selected classes were used while training the models. To prevent overfitting, geometric data augmentation methods such as sampling, mirroring, rotating, and shifting were applied to the original dataset. To test the models, 100000 items per class were chosen, and the data were split into three different folds: 70% for training, 15% for validation, and 15% for testing. An object can be drawn in many ways, which causes divergent sketches for the same object category. We applied t-SNE on selected classes to visualizing high-dimensional data in a two-dimensional map. For further similarity search, the k-Nearest Neighbors (kNN) algorithm is utilized.

2.3 Algorithms and Networks

The study includes two main networks, RNN and CNN, and other classification algorithms and networks to gain insights into the data and evaluate methods.

The Random Forest [3] algorithm works by creating many decision trees randomly and use them as predictors or classifiers. By combining individual tree votes, it makes the ultimate decision that provides resistance to overfitting. We also evaluated the Support Vector Machines (SVM) [2], which divides the data into different classes by creating a line or hyper-plane.

The first neural network tested is a Multi-Layer Perceptron (MLP) which consists of two hidden layers with sizes of 512 and 256, and the learning rate as 0.001. Also, a Rectifier Linear Unit Activation (ReLU) function and a Cross-Entropy loss function have been used. Standard RNNs have problems with long-term dependencies. Therefore, different versions of RNNs have been developed to solve the vanishing gradient problem. The Long Short Term Memory (LSTM) is a modified version of RNNs explicitly designed to solve this problem [9]. In this implementation, the first model was a sequential CNN, and as a second model, bidirectional LSTM was used. In addition to those networks, initial accuracy comparison tests on the dataset were conducted using ResNet, a deep residual network proposed by He et al. [7] and standard module of MobileNet in Keras. According to the experiments conducted with the selected group of objects, the speed requirements of real-time response and adaptability to the game engine demand simpler network architectures.

The structure of the networks and their parameters were chosen based on the results that provided fast predictions through the game. In the LSTM network design, a unit with two layers was used for the nonlinear mapping, and a fully connected layer followed by a sigmoid function for the nonlinear function that projects the hidden states to the outputs was used. The final CNN network consists of three convolutional layers, each followed by a max-pooling layer, complemented by a flatten layer and two dense layers. All convolutional layers employ a ReLU activation function, whereas the first dense layer uses the tanh function and the second dense layer uses the softmax function. Adaptive Moments (ADAM) was used as an optimizer, and the libraries Keras and TensorFlow were used.

2.4 Game Implementation

The Barracuda package [10], a lightweight cross-platform neural network inference library for Unity that can run neural networks, was used to deploy the trained model into Unity. Selected CNN model on selected classes of Quick Draw dataset is converted to ONNX format and uploaded to Unity. Models of the objects are loaded to the game engine as assets, and prefabs are called when the player's sketches are drawn and submitted. The similarity percentages of all classes are listed as an array, and the code checks the highest accuracy result. As soon as the player starts to draw, received strokes from the canvas are analyzed and sent to the model to recognize the player's sketch. When the player hits the submit button, if the drawing has a similarity result above the identified percentage, necessary operations are executed according to the object type. If the similarity is below that percentage, a warning is given. After the tests, the threshold for similarity results was identified as 75%.

2.5 User-Experience Tests

The crucial factor for the implementation evaluation of sketching as a game mechanic is the involvement of potential users. Therefore, a user test was conducted with 18 participants (nine male and nine female). Recruited respondents were between the ages of 19–29, with a mean age of 22.52 years ($SD = 4.13$). Subjects have been notified and properly instructed about their voluntary participation and the experiment procedure. Ethical Approval of Research was granted by the Middle East Technical University Human Subjects Ethics Committee in May 2021. The questionnaires were adapted from the literature. Participants were first asked to evaluate sketching as an interaction medium for games on a 5-point Likert scale. After playing the game using a computer, four questionnaires, Gamer Profile [5], System Usability Scale (SUS) [4], Technology Acceptance Model (TAM) [11], and GAMEFULQUEST [8], were collected. TAM questionnaire covered descriptive questions on a 10-point scale to determine the level of technology acceptance. SUS is a psychometric tool that takes into consideration three usability criteria: effectiveness, efficiency, and satisfaction. There are ten questions in the SUS that score on a 5-point Likert scale, where odd-numbered questions consist of positive statements, whereas even-numbered ones include negative statements. GAMEFULQUEST [8] is initially proposed for gamification context as a guideline to evaluate gameful experiences. To evaluate the multidimensional nature of game experiences and dimensions, GAMEFULQUEST is used. Participants played the game for 15 minutes, and the total process took 30 minutes for each participant.

3 Results and Discussion

3.1 Technical Results

To obtain higher accuracy and better performance, we designed different networks as explained above and tested various algorithms. They were tested for their performances by changing the structure and parameters to determine the most suitable method for real-time interaction. Based on the performance on a validation set, the training processes were terminated when the performance stops improving (Table 1) to avoid over-training and overfitting. While adding 0.1 as the dropout value improved the accuracy of CNN, using 0.3 as the dropout value improved the results of the LSTM model (Table 2). The regularization effect of dropout reduced the overfitting; consequently, dropout layers were added. The learning rate was defined as 0.01 for CNN and 0.0001 for the LSTM (Table 3).

A timer was added to the game engine to calculate the response time, which stops when the model unfolds the final accuracy. To determine the model used for the game, besides the accuracy and timing aspects, precision, recall, and F1 score through all tracks in the test data were calculated to evaluate the effectiveness of each model (Table 4). We fed the same nine classes of sketches to the sketch classifiers for quantitative evaluation. Each network was trained for a maximum

Table 1. Epoch and Accuracy

| Epoch | CNN | LSTM |
|-----------|---------------|---------------|
| 5 | 90.42% | 89.51% |
| 10 | 94.62% | 91.65% |
| 15 | 95.50% | 92.52% |
| 20 | 96.10% | 92.50% |
| 25 | 96.32% | 92.89% |
| 30 | 96.02% | 92.93% |
| 50 | 96.25% | 92.65% |

Table 2. Dropout and Accuracy

| Dropout | CNN | LSTM |
|------------|---------------|---------------|
| - | 95.62% | 91.80% |
| 0.1 | 96.32% | 92.68% |
| 0.15 | 95.52% | 92.10% |
| 0.2 | 96.00% | 92.40% |
| 0.3 | 94.00% | 92.93% |
| 0.4 | 96.32% | 90.25% |
| 0.5 | 96.32% | 88.90% |

Table 3. Learning Rate and Accuracy

| Learning Rate | CNN | LSTM |
|---------------|---------------|---------------|
| 0.000075 | 40.32% | 88.00% |
| 0.0001 | 80.32% | 92.93% |
| 0.001 | 86.40% | 30.26% |
| 0.005 | 90.65% | 25.00% |
| 0.0075 | 93.32% | 17.50% |
| 0.01 | 96.32% | 14.13% |
| 0.05 | 50.32% | 9.45% |
| 0.1 | 35.20% | 8.00% |

Table 4. Statistical Analysis of Models

| | CNN | ResNet | LSTM |
|---------------|----------|---------|---------|
| Accuracy | 96.32% | 93.00% | 92.93 % |
| Recall | 0.91 | 0.92 | 0.91 |
| Precision | 0.90 | 0.92 | 0.92 |
| F1 score | 0.90 | 0.92 | 0.91 |
| Reaction Time | 0.70 (s) | 1.9 (s) | 0.97(s) |

of 50 epochs. The results are summarized in Table 5. The performances of the two architectures, ResNet and MobileNet, which have already reached state-of-the-art levels for image classification, still came short compared to CNN.

Table 5. Accuracy Results

| Network / Algorithm | LSTM | CNN | SVM | kNN | RF | MLP | MobileNet | ResNet |
|---------------------|--------|--------|--------|--------|--------|--------|-----------|--------|
| Accuracy | 92.93% | 96.32% | 87.21% | 90.57% | 89.17% | 90.24% | 94.10% | 93.00% |

After those processes, the final CNN model with 96.32% recognition accuracy, 0.7s reaction time, and 0.91 F1 scores could immediately respond to the player without a noticeable delay. Due to the performance of LSTM in previous studies [1], we expected higher accuracy results from LSTM. The reason behind these results can be random properties stemming from the temporal information. LSTM learns the temporal evolution of the data by preserving sequential information. Therefore, selected classes might have affected the results due to the nature of the sketches. Also, according to [1], an increase in the number of samples resulted in the success of the LSTM. Sharing the same core idea that works in LSTM units, ResNet showed similar performance accuracy for a small dataset. Therefore, if a larger dataset is required, LSTM or ResNet can be preferable due to their previous successes on large datasets.

3.2 User-Test Results

The user-test results of the pre-test ($M = 2.72$, $SD = 1.28$) and post-test ($M = 4.5$, $SD = 0.6$) indicate substantial increase. According to two-tailed t-test, p-value is less than 0.0001 ($t(18) = 7.5179$), demonstrating an extremely statistically significant outcome. The results show that game Sketchscape significantly changes participants' ideas on sketching in a game. The majority of the participants prefer to use full-screen canvas, indicating that it felt more natural (Side Panel: 33.3%, Full Screen: 67.7%). Cronbach's Alpha value was calculated as 0.81, which indicates a high-reliability level. Figure 2 indicates the SUS score of the game is 82.1, which states "good" usability. Results of the GAME-FULQUEST are presented in Figure 3. Participants indicated they felt accomplishment when they solved the problems or found new ways to use spawned objects. Most of the participants were willing to improve their methods. The majority of the participants reported that the game creates internal competitiveness and requires creativity.

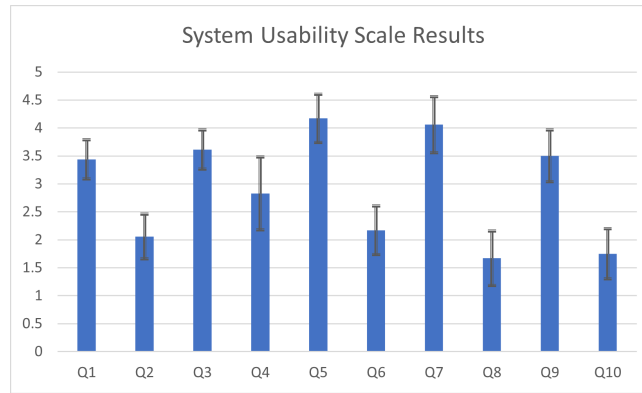


Fig. 2. System Usability Scale Results.

4 Conclusion and Future Work

This study presents a novel interaction method for interactive gameplay that uses sketch recognition to enable the players to interact with the narrative by drawing objects. A prototype maze game is developed to demonstrate the implementation of the model and create responsive and interactive game experiences. In the experiments, the proposed game mechanics and the deep learning model reveal promising results. The user study results and participants' preferences direct us to increase the classes and enrich the interaction modes. As future work, we intend to combine this study with sketch-based 3D model retrieval methods. This method will aim to find the most relevant visual model, and due to the

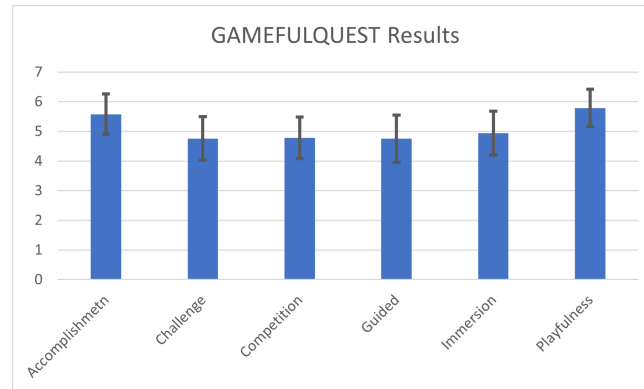


Fig. 3. GAMEFULQUEST Results.

different features of the models belonging to the same object, players will create new directions in the narrative.

References

1. Andersson, M., Maja, A., Hedar, S.: Sketch classification with neural networks: A comparative study of cnn and rnn on the quick, draw! data set (2018)
2. Awad, M., Khanna, R.: Support vector machines for classification. In: *Efficient Learning Machines*, pp. 39–66. Springer (2015)
3. Breiman, L.: Random forests. *Machine learning* **45**(1), 5–32 (2001)
4. Brooke, J.: Sus: a “quick and dirty” usability. *Usability evaluation in industry* **189** (1996)
5. Foundry, Q.: Gamer motivation model. <https://apps.quantificfoundry.com/surveys/start/gamerprofile/> (2016)
6. Google: Quick,draw (Nov 2016), <https://quickdraw.withgoogle.com/>
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 770–778 (2016)
8. Högberg, J., Hamari, J., Wästlund, E.: Gameful experience questionnaire (gamefulquest): an instrument for measuring the perceived gamefulness of system use. *User Modeling and User-Adapted Interaction* **29**(3), 619–660 (2019)
9. Sherstinsky, A.: Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena* **404**, 132306 (2020)
10. Unity: Unity barracuda (Nov 2019), <https://docs.unity3d.com/Packages/com.unity.barracuda@0.3/manual/index.html>
11. Venkatesh, V., Davis, F.: A model of the antecedents of perceived ease of use: Development and test*. *Decision Sciences* **27**, 451 – 481 (06 2007). <https://doi.org/10.1111/j.1540-5915.1996.tb00860.x>
12. Williford, B., Runyon, M., Malla, A., Li, W., Linsey, J., Hammond, T.: Zensketch: A sketch-based game for improving line work. pp. 591–598 (October 2017). <https://doi.org/10.1145/3130859.3130861>