A.I, GO DOODLE: Web Game Service with A.I

Dongwoo Kim[1], Donghan Kim[2], Yeonghyeon Kim[1], and Hyeonggyu Choi[3]

[1] Sungkyunkwan University, Mathematics
[2] Sungkyunkwan University, French Language and Literature
[3] Sungkyunkwan University, Integrative Biotechnology

Abstract. A.I. GO DOODLE is a web game in which up to four people participate, draw and enjoy with given random keywords. In addition, artificial intelligence intervenes to score sketches drawn by players or to guess the label of the sketches. For this role, image classification or object detection technology will be applied with A.I. according to our resources. We planned to train A.I. with subset of a "Quick, Draw! The data" open-source dataset which has 345 categories of 500,000 sketch data. We also planned to use socket.io to build a system so that multiple players can participate and chat at the same time.

Keywords: A.I,GO DOODLE · Web Game Service · Artificial Intelligence · Drawing

1 Introduction

'Covid-19' will be the keyword that has dominated social issues in recent years. This virus changed our lives so much. As we can see in the phrase 'stay home, save lives', the time spent at home has increased and communication with people has decreased significantly. Web game "A.I. GO DOODLE" is designed for people who lack such entertainment and are bored with quarantine. We developed a game for many people under the motto of easy, simple, and enjoyable. In addition, we decided to introduce AI that will play a role as a scorer or examiner of the game. With the development of artificial intelligence technology, since people's interest has increased. By using this technology in our games, we hope players to feel that A.I. is not a difficult thing, but it is familiar and fun.

2 Objective

Our project aims to create a novel web-based game service with following features. First, a deep learning model trained with the doodle dataset is applied to the game to help users perceive AI as familiar and interesting. Second, it enables an immediate reaction with the user through AI's fast and accurate picture recognition. Third, it stimulates users' interest through a simple UI and a fun Game Mode. Fourth, multiple users can chat and draw pictures at the same time to communicate with each other and have fun.

3 Related Work

3.1 Related games

3.1.1 Catch mind ‘Catch Mind’ is a computer game serviced by Netmarble. In a game of ‘Catch Mind’, Each player takes turns and draws a picture that matches the problem. And if someone else gets the correct answer on the picture, both the person who drew it and the person who guessed it get experience points and credits. The game is played for 20 rounds, and the team or user with the most total points wins.

3.1.2 Quick, Draw! ‘Quick, Draw!’ is an online game developed by Google that challenges the players to draw a picture of an object or idea for a given keyword and then uses artificial intelligence to guess what the drawings represent. The game is similar to ‘Pictionary’ in that the player only has a limited time to draw. In a game of ‘Quick, Draw!’, there are six rounds. Before each round, the player is given 20 seconds to draw a randomized object selected from the game’s database. As they begin each round, they have 20 seconds to draw the given prompt, whilst the artificial intelligence attempts to guess the drawing. A round ends either when the artificial intelligence successfully guesses the drawing or the player runs out of time. At the end of a ‘Quick, Draw!’ game, the player is given their drawing and results of each rounds. The player can also check the artificial intelligence’s comparison and guesses with other player-given objects, before either quitting or replaying.

3.1.3 Gartic Phone ‘Gartic Phone’ is a web game released in December 2020 by Brazilian studio Onrizon. In a game of ‘Gartic Phone’, when the game starts, each player basically starts with their own first sentence or first picture, depending on the mode. Then, on the next turn, he sees the first result of the other person and continues with a sentence or picture according to his inferred content. In this way, the next turn also takes a method of looking at other people’s previous results and continuing. If you complete the last turn in that way, you will have time to appreciate the album that contains the process so far. The point is to go through the abstract drawings and sentences of the participants in this process, and see how amazingly they change from the initial theme.

3.1.4 Cardgames.io ‘Cardgames.io’ is a website that collects web implementations of various board games. We focused on the accessibility of the games on this site. Many existing games require an installation and a login process. These are obstacle for users who want to enjoy a simple game with their friends. On the other hand, io games do not require this process. Users can enjoy multiplayer mode by choosing their nickname to use temporarily and creating a room simply. Since all of the resources of the game exists on the web page, and its simplicity, players can enjoy it right after they access the site. We were fascinated by this accessibility of cardgame.io and served as a motivation for us to plan A.I. GO DOODLE as a web game.

3.2 Socket.io

Since our game operates in real-time in Web, we decided to use socket.io library. Socket.IO is a library that enables real-time, bidirectional and event-based communication between the browser and the server. It is built on top of the WebSocket protocol and provides additional guarantees like fallback to HTTP long-polling or automatic reconnection.

4 Proposed Solution

4.1 Overall Architecture

Web pages for game will be created with pug template engine and node express web server. We have two main server for this project, node.js server for game engine and and simple python server for A.I works and http response. node.js server will render game and connects users in real-time using Web socket. python server will receive image file that is drawn in real-time by users as JSON file format and then return the accuracy of images as result to game server. In this project, these servers communicate each other using REST API.

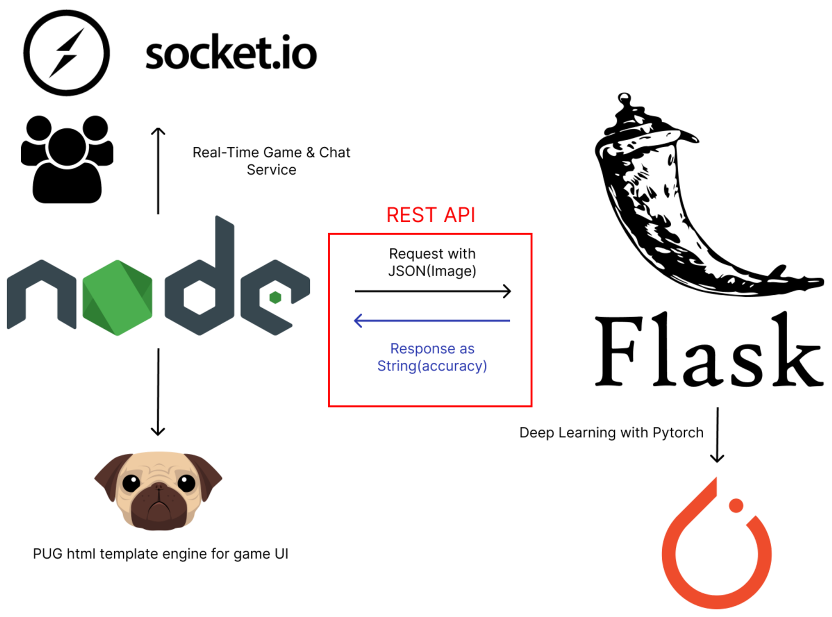

socket.io
Real-Time Game & Chat Service
REST API
Request with JSON(Image)
Response as String(accuracy)
node
Flask
Deep Learning with Pytorch
PUG html template engine for game UI

Fig. 1. Overall Architecture

4.2 Servers Workflow

A game server created with node.js and express frameworks delivers html files that implement the game UI to users who access the game. And then, when the game starts, users in the same room are tied up through socket communication using the socket.io package to maintain sessions and enable real-time chat and game operation. When users finish sketching each game, the game server send a post request with an image file attached in JSON file format to the flask server for AI operation. The flask server for AI work interprets the JSON file contained in the request sent from the game server as an image file, and evaluates the accuracy value of each image after deep learning using phytorch. These evaluated values are sent back to the game server in the form of strings. The game server then processes the responses received from the Flask server and outputs the results to the users.

4.3 Applying AI

Once the sketch data of each user are handed over to the database, the grading model grades and scores each of them, independently. Since the score of the sketch is based on 'how good does the user's sketch depicts the given keyword', the accuracy of the grading model is important. We used CNN-based deep learning model since they showed a great performance on classifying the image data. Here, we propose two methods for grading the user's sketch.

4.3.1 Image Classification Image Classification is one of the core problem in Computer Vision, which aims to predict the class of the object in the image. Until the 2011, the feature descriptor based methods, which analyzes the feature of the given image and predicts the class, was superior to any other methods. However, in 2012, the AlexNet[1], which took first place on 2012 ImageNet Large Scale Visual Recognition Challenge (ILSVRC), showed outstanding performance of the deep convolutional neural network on image classification task. This triggered various research on architecture of the model, appropriate hyperparameters, activation functions, etc. We aim to find the model that shows the highest accuracy on our dataset. The current state-of-the-art model on image classification is CoAtNet[2], but we excluded it from our model candidate since it has too many parameters, which can possibly cause overfitting on our dataset. Considering our limitation of computational resources, we propose to deal our image classification task with three candidate models: VggNet16 and 19[3], Resnet-50[4] and mobilenet[5].

4.3.2 Object Detection It seems that the image classification models can manage our grading system reasonably, but there is one inherent problem: the Image classification model assumes that there exists exactly one class per one image. So, just applying image classification model would perform poorly if there are multi-objects in the user's image. It has not been decided whether our game to provide one keyword or several keywords, but it would be advantageous to take measures. So we also suggest object detection as a grading system of our game. Object detection algorithms can be classified into two categories: two-stage or one-stage models. Two-stage model, which is also called multi-stage model, extracts region of objects and then classify the object. These stages are called localization and object classification, respectively. This method shows high accuracy, but low speed due to its separated task handling. One-stage model, which is also called single-stage model, runs detection over a dense sampling of locations directly. Although one-stage shows lower accuracy compared to the previous model, it has high speed due to its concurrent localization and object classification task. We propose faster R-CNN[6] and Yolo v1[7] as our object detection candidate model, which are two-stage and one-stage model respectively.

4.3.3 Dataset Since our grading system is based on machine-learning algorithm, we need appropriate dataset to train the model. The dataset we require must meet the following conditions.

- sufficient amount of sketch image data for training
- more than 30 categories
- open source data which can be used without any restriction Fortunately, the quick draw open source dataset[8], which is provided by google, met our requirements.

The quick draw dataset has more than 50 million drawings across 345 categories, and it is shared in an open source manner. The preprocessed data is provided in various formats such as ndjason, bin or npy. Below we plotted some drawing data of npy format as an example.

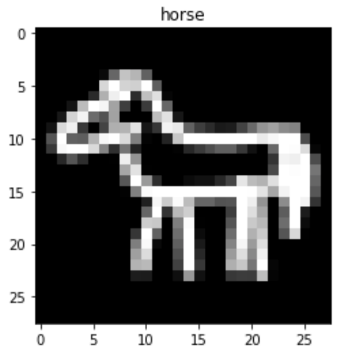


Fig. 2. horse npy

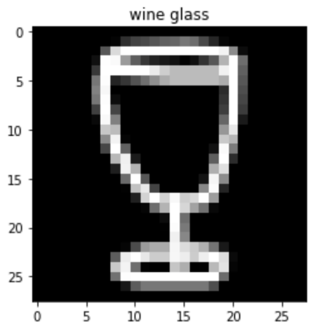
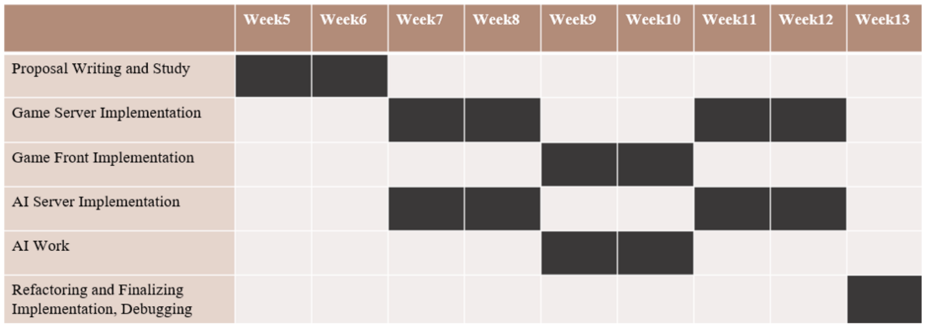


Fig. 3. Wine Glass npy

5 Planning in Detail



	Week5	Week6	Week7	Week8	Week9	Week10	Week11	Week12	Week13
Proposal Writing and Study	■	■							
Game Server Implementation			■	■			■	■	
Game Front Implementation					■	■			
AI Server Implementation			■	■			■	■	
AI Work					■	■			
Refactoring and Finalizing Implementation, Debugging									■

Fig. 4. Detail Plan Table

We were divided into two teams, each in charge of building the game environment and AI, and decided to help each other after the study for each team. After each server is built in 7-8 weeks, and the main technologies are implemented in 9-10 weeks, everyone gathers together to complete the overall process, and then finalizes it through testing and refactoring in 13 weeks.

References

1. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.
2. Zihang Dai, Hanxiao Liu, Quoc Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. *Advances in Neural Information Processing Systems*, 34, 2021.
3. Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
4. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
5. Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.
6. Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.
7. Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.
8. google. Quick, draw! the data. https://quickdraw.withgoogle.com/data/. [Online; accessed 25-Mar-2022].