

Draw This! – An Exploration of the Mind and Its Interaction with Mobile Technology

Andy Hoover
Computer Science & Mathematics
College of Wooster
ahoover11@gmail.com

Ronald Lopez
Computer Science
Kean University
Ronlp123@gmail.com

Webb Chawla
Computer Science
North Carolina State University
Vchawla3@ncsu.edu

Ignacio Dominguez
Computer Science
North Carolina State University
ignacioxd@ncsu.edu

David Roberts
Computer Science
North Carolina State University
robertsd@csc.ncsu.edu

Abstract

The purpose of this project is to explore the way people interact with mobile devices. An Android application entitled “Draw This” was developed, which is a single player game that requires a user to draw a specific object when prompted with either an image or text. Players receive a score for their drawing based on how similar it is to a selected reference image. The score for a user’s drawing is generated by inputting the drawing into a neural network that has been trained to recognize the specific object. The score of the drawing will be based on the activation level of the neuron pertaining to the selected reference image. During each play session, player interactions are logged, which includes information such as accelerometer readings, buttons clicked, touch events, and drawing patterns. Analyzing the user’s patterns of game play will allow us to connect users’ cognitive processes to their interactions within the application.

Introduction

The research project is based on a drawing game developed in Android called Draw This. Draw This is a mobile app similar to Draw Something, a multiplayer social drawing game developed by OMGPop and later bought by Zynaga for the iOS platform. Instead, Draw This is a single player game where users will be presented with an object at the beginning of the game. The object can either be a text or an image. In the game, the user will draw the object and their drawing will automatically be scored. The application was written in Java and developed in a beta version of Android Studio which is a new integrated development environment (IDE) for the Android platform.

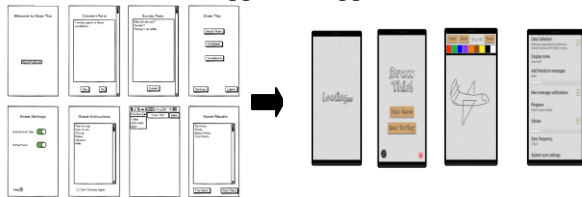


Development

The tasks for the project were divided between 2 team members and me with the guidance of our mentor and his assistant. The tasks were to work on survey form, consent form, layout flow, screenshots, data collection, activity design, game screen, app settings, and the scoring algorithm. My role was to work on the layout flow of the game like how the entire game will look and be played out. Then, design some activities of the game which in my case was to create the loading page, the main menu, and settings. Next, implement the app settings for the game so users can select their preferences on how they want to play the game. Finally, work on the scoring algorithm of the game for players to earn a score for their drawing.

For the layout flow of the game, I used mockups to help view the way the game should be or at least start off. These mockups go page by page when the user enters the game. Then after using the mockups as a guidance I was prepared to begin the actual design and implementation of the activities [1]. It began with a loading page when the app was tapped to be launched. A loading text will appear for two to three seconds depending on the speed of the device and open the consent form. The consent form will be displayed with a yes or no button if people agree to the consent or not. If the user would select yes, it will open the survey form for questionnaire and move on to the main menu. If they select no, it will take them straight to the main menu of the game. At the Main Menu screen, the activity contains the title of the game, two buttons to either start a new game or learn how to play, and two more buttons for settings and quit. The title of the game is a text with a customized font. The “New Game” button starts the game screen page developed by my colleague, Andy Hoover. The “How to Play” button opens an instruction page of how to play the game. The setting button is an image of a gear which substitutes the plain button for a

gaming style one. When the setting button is tapped it will open the application settings for the game. At the same time, I program the app settings with the assistant of the Android documentation. The most important option I implemented was a checkbox for the consent form so if someone wanted to change their mind about their data being collected, they can check the box to choose their preference. The quit button is an image of an exit symbol and also used for a gaming style one instead of a plain button. When this button is tapped the application will close.



After finishing the game design, it was time to solve the scoring algorithm for the game. The scoring algorithm was the challenging portion of the application. Why do we want players to earn a score? A score determines a reason to play the game because without it users wouldn't enjoy playing the game. They wouldn't enjoy playing the game because there wouldn't be a point drawing an image and getting nothing in return. We want players to have fun and to keep playing so we can collect more data from more users. The score was the key to making this application a successful gaming app and attract more gamers for data collection.

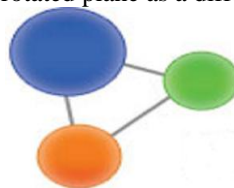
Hypothesis

How can players earn a score for their drawing? The idea was to compare the user's drawing to an existing drawing and earn points based on how similar they are. The search was on for what tools can be used to help this process be successful. The tools discovered were OpenCV and Neuroph Studio and both of them were related to image processing.

OpenCV is a library of programming functions for computer vision applications developed by Intel. It focuses on real-time image processing [2]. This tool was used for feature matching between the drawing's features and reference image's features [3]. For example an airplane's wing and tail. The error when testing this tool was the drawings the player made were too simplistic. Also the reference image that was being used to compare was too detailed like color, design, and sharpness. Even comparing two basic drawings, it failed to tell if the drawings had any actual features. In the end, OpenCV was used for better quality images so it can read the features of the object properly rather than a simple drawing made from a finger.



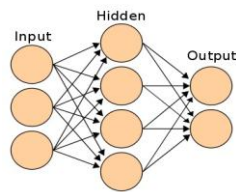
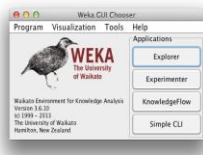
Neuroph Studio is an object-oriented neural network framework written in Java [4]. Its purpose is to create and train neural networks. Neuroph Studio was used for image recognition to see if it can recognize the drawing based on the similar images that were fed to the neural network and trained with [5]. The problem using this tool was the GUI was too complicated. It was hard to understand all of the unnecessary functions and controls that had to be used to train the neural network. The GUI was too slow and eventually crash every thirty minutes. The system also didn't allow the classification of multiple images as the same object. For example if the object is a plane and then we rotate that plane it should still be the same object but instead it was classifying the rotated plane as a different object.



Current Status

As my colleague, Andy Hoover, finished their part of the project he began to assist me with the scoring algorithm for the game. Our mentor, Professor David Roberts, mentioned a useful tool that will help solve the scoring algorithm. The tool was called Weka and he talked about how we can use it for classification. The idea was to make a classifier built with many images of the same object and using that classifier to classify the user's drawing [7]. In definition, Weka is a collection of machine learning software. It was developed at the University of Waikato, New Zealand and written in Java. Weka contains a large set of tools for data pre-processing, classification, visualization, and more [6]. With this tool in hand we were able to do what Neuroph Studio couldn't do which was classifying multiple images as the same object. It was complex but relatively easy to use. Therefore creating and training the neural networks were simple and fast to do. The artificial neural networks (ANNs) are presented as a system of interconnected neurons which can compute values from inputs [8]. The input layer of the system was where the data will be introduced. The hidden layer was where the processing occurs through the system of weighted connections. Then, the output layer

would output the answer. Using this model helped us understand how the neural networks works and looks. Therefore, when we tested Weka to recognize an image it worked and gave us a value of 92%. From here, Weka was our classifier!



Results

There was no data to be recorded due to the incompleteness of the application. The research will still be executed and recorded for future missions from other students.

Conclusion & Next Step

The drawing game, Draw This, is currently about 85% done. The missing 15% is the scoring algorithm as well as minor functions that have to be added and fixed. For the scoring algorithm, the objective is to generate models from Weka for each type of object which can be used to calculate a score for an image when inputted into the correct model, where the score is based on the activation level of the neuron pertaining to the selected reference image. The next step is to collect data throughout the entire game when the user launches the game, plays the game, and quits the game. The data collected are the scores the player earned for their drawing, timestamps,

game play, and drawing patterns. Analyzing this collection of data that people go through will make it easier for us to determine the various cognitive processes that the minds go through when humans interact with mobile devices and applications. The future goal is to release the app in the Google Play Market to collect more data from users throughout the country.



References

- [1] Android Documentation. Introduction to Android. Available: <https://developer.android.com/guide/index.html>
- [2] OpenCV Documentation. OpenCV 2.4.9 (2014-04-25). Available: <http://opencv.org/documentation.html>
- [3] OpenCV in Android. Introduction into Android Development. Available: http://docs.opencv.org/doc/tutorials/introduction/android_binary_package/android_dev_intro.html
- [4] Neuroph Documentation. Sourceforge. Available: <http://neuroph.sourceforge.net/documentation.html>
- [5] Neuroph for Android. Creating Android Image Recognition Application using Netbeans and Neuroph. By: Dalibor Micic. Available: http://neuroph.sourceforge.net/tutorials/android_image_recognition_using_neuroph.htm
- [6] Weka Documentation. Developers: Eibe Frank, Mark Hall, Peter Reutemann, and Len Trigg The University of Waikato. Available: <http://www.cs.waikato.ac.nz/ml/weka/documentation.html>
- [7] Weka in Java. Use Weka in your Java code. Available: <http://weka.wikispaces.com/Use+WEKA+in+your+Java+code>
- [8] Definition of Neural Networks. Artificial Neural Networks. Wikipedia. Available: http://en.wikipedia.org/wiki/Artificial_neural_network