Generation of 3D objects based on Virtual Reality drawings using Convolutional Neural Networks

Wilson Araujo de Oliveira Neto, Kid Mendes de Oliveira Neto, Jucimar Maia da Silva Junior
Samsung Ocean Center (OCEAN)
Universidade do Estado do Amazonas
Manaus, Brazil
wadon.snf@uea.edu.br, kmdon.snf16@uea.edu.br, jjunior@uea.edu.br

***Abstract*—Despite the recent advances in virtual reality devices and the increasing availability of the equipment on the market, there are still challenges to bring these technologies to all kinds of public, mainly due to the absence of a standard graphics interface in the virtual reality environment development. Thus, this paper aims to propose the use of interaction in virtual reality integrating convolutional neural networks, attempting to improve the experience by offering an easily understandable and useful environment.**

***Keywords*-Virtual reality; neural networks; drawing; convolutional neural networks; machine learning;human computer interaction;**

I. Introduction

The virtual reality has been originated in the decade of 1950, with the experience of the multisensorial immersive machine named Sensorama. Created by Morton Heilig, the machine was an attempt to integrate all the body senses in an efficient way in a pre-recorded motorcycle ride in Manhattan, providing to the user an engagement of the body senses: hearing, vision, smell. Although the area has had its first experiments in the 1950s, was carried out in the 1990s, with high-quality technologies and performance in the refinement of interaction the most equivalent possible to the sensory senses of the human body.

Due to the advances in technology and the decreasing costs of production, the access to an immersive virtual reality environment has grown resulting in the creation of several products, for example, the use of smartphones as displays, making possible the creation and the distribution of HMDs (Head Mounted-Displays) for domestic use of virtual reality.

For the operation of the system, three concepts are required: immersion, interaction, and involvement [1]. The importance of those concepts is given in order to achieve stimulate the greater quantities of senses, aiming to have a greater realism. In spite of all these conceptions, a pattern has not been found to be able to follow and mirror itself by providing the best immersion in a virtual environment. As a way of providing a solution, the VR area relies on other areas, especially in the use of Machine Learning.

The artificial intelligence field has many branches, including machine learning, which has in the latest years increased its accuracy in specific projects, thanks to the use of neural networks. The classification and object detection in images has become fundamental in the field of computer vision [2]. Among current techniques for image recognition is the use of convolutional neural networks (CNN), which has become the state of art by performing classifications with few layers and a much lower cost than simple networks.

In this context, the main contribution of this paper is to propose the use of CNN combined with VR to improve the interactions of the users with the virtual interfaces making it easier and more fun to access or invoke elements in applications through the drawing.

This work is organized as follows: Section 2 points out the main works related to this article, Section 3 describes the proposed system, Section 4 presents the experiments and results obtained, and finally, section 5 presents the conclusions and possibilities of future works.

II. Related Works

There are several methods of image recognition for 3D objects creation, one of those uses a CAD(Computer-Aided Design) file which is scanned to generate a 3D model. According to [3] recognition processes are not fully vectored and can display errors in image recognition.

The introduction of a new system to assist designers in their personal production integrating the use of mixed reality MixFab is designed to reduce the 3D content, from casual designers, producing an artificial model with their own hands, replacing the need of an advanced modeling ability with intuitive gestures [4].

MixFab provides an Augmented Reality environment, in which users can interact naturally without the need for equipment. However, the image recognition does not represent the standard size, causing resizing and automation problems. In the proposed software the data is inputted by a Joystick, as the user moves it the drawing is made with more precision and success, thanks the image recognition performed by the neural networks.

III. Development

The drawing recognition system was divided into 2 modules, as can be seen in the Fig. 1, the first one (Server) was

developed in Python language and the second one using the Unity 3D Engine and C#.

In order for the game works with minimum delay as possible, it was necessary to train the predictive model before the game started. In favor of having a good hit rate on the model, it requires a lot of data training and a lot of time. The trained model allows machines with older hardware configurations to run the server and classify the images it received.

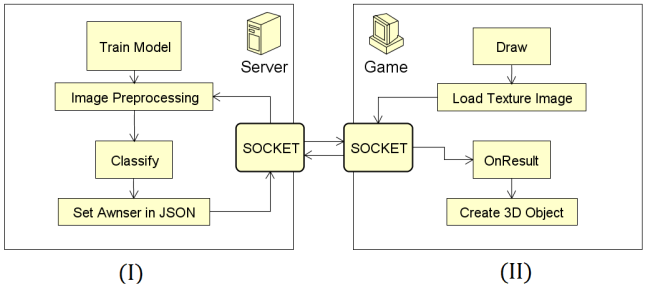

Figure 1. Proposed System.

The training of the model was done from a database and then made available for consumption in WebSocket server format (I). After the game connects to the server, the application sends the image drawn to the server and receives as a response the category that should be a 3D rendered (II). It may be a baseball ball or a baseball bat.

A. Client-Server Communication

Socket.io 2.0 was used with the Python 3.6 language, making possible the bi-directional communication between Client-Server. This module is a library made to build real-time applications, which uses WebSockets specifications and defines an API that establishes 'socket' connections between a client and a server in a persistent way, allowing the sending and reception of data at any time.

Socket.io opens a communication channel between the server and the client persistently, similar to a chat room. The client sends to the server the image in text format through the base64 encoding, after processing the server, receives a message through the same channel containing the identification of the 3D object to be rendered.

B. Digital Picture

For the creation of the digital picture, we used the HMD Odyssey Glasses and the HMD Odyssey Controllers. Allowing the detections of the movements with the hands, aiming to provide an immersive experience when drawing, according to the Fig. 2.

At the end of the draw mode, the digital picture is stored in a texture and then a function will be started to capture the rendering of the texture, immediately saving it in a PNG image and sending it to the Server.

Figure 2. HMD Odyssey Glasses and HMD Odyssey Controllers.

C. Model Training

The neural network was used to classify the images received via WebSocket. The training was done with $241,763$ images from the Google repository called *quickdraw*, being $115,324$ for baseball bats images and $126,439$ for baseball balls images. The processing consists in normalizing the pixel data from 0 to 1 to reduce the error rate [5], in addition to reducing the processing time, the dimensions of the images were reduced to 28 x 28 pixels. The neural network has two outputs represented by the percentage of each category.

The architecture of the neural network is defined in the Fig. 3. It was used $1,554,694$ trainable parameters and 85 times to get an accuracy of 95% according to the validation of 30% of the original dataset i.e. $72,529$ elements.

layer name	output size	layers
conv1	11×11	7×7, 64, stride 2
conv2_x	5×5	3×3 max pool, stride 2
		[3×3, 64; 3×3, 64] ×2 [3×3, 64; 3×3, 64] ×3 [1×1, 64; 3×3, 64; 1×1, 256] ×3
		[1×1, 64; 3×3, 64; 1×1, 256] ×3 [1×1, 64; 3×3, 64; 1×1, 256] ×3
conv3_x	3×3	[3×3, 128; 3×3, 128] ×2 [3×3, 128; 3×3, 128] ×4
	1×1	average pool, 1000-d fc, softmax

Figure 3. Neural network architecture.

The trained model is saved to a file so that it can be consumed by the server later. This model is a set of neural network architecture and weights.

D. Image Recognition

For image recognition, it is necessary to make some preprocessing steps, so that it is similar to the dataset. First,

the image is received in text type in base64 and then decoded to the Image data type and converted to grayscale. Then resized to the size 28x28 pixels. Finally, the colors are reversed so that the color pattern that will express the content is only in the traces and not in the background of the image, according to the Fig 4.

It is possible to predict the class from the image. However, only text in JSON format containing the object code will be rendered to the game.

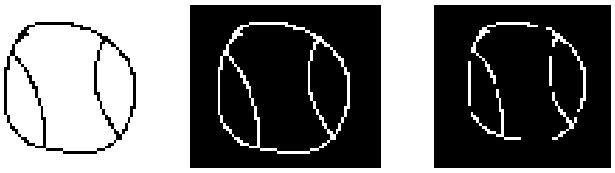

Figure 4. Image preprocessing.

IV. Experiments and Results

In this section, CNN's training and experiment environment will be presented. The following resources were used: 2 Notebooks with Intel i7, 8 GB of memory RAM, GEFORCE GTX 1050 graphics card with 4 GB of memory, Windows Operating System, HMD Odyssey, scientific computing framework Keras and an engine to create games Unity3D.

A. Experimental Procedure

The experiments aimed to simulate a real situation of the game, in which a player uses a computer that communicates via WebSocket with the server. To test, several balls and baseballs bats objects were drawn, as can be seen in the Fig. 5, the drawn images have their respective classifications by the neural network.

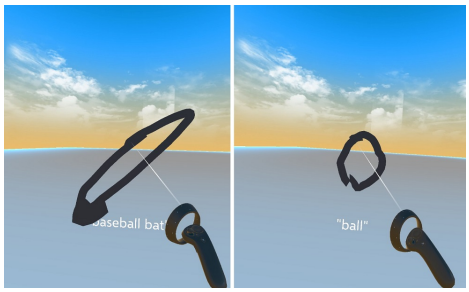

Figure 5. Drawn images test.

The experiment was conducted with 6 people in pairs. The spectator was responsible for evaluating the result of the drawing and making sure the model was hitting or not. Each user made 10 draws from each class, totaling 120 and based on them, the real-world accuracy of the application was calculated.

B. Results

Given the result of the neural network, it became possible to create 3D elements that correspond to the image drawn as, for example, balls and baseballs bats, observed in the Fig. 6. It was observed that the interaction of the user with the game became fluid and natural from the tests, facilitating the accuracy of the classification model. During the tests, it was noticed that communication with the server occurred in real time, and it was not possible to distinguish if the predictive model was implemented in the game itself.

Figure 6. 3D object creation.

The evaluation metrics according to Table 1, explain the accuracy of the model with real tests. The values represented are P for the baseball bat and N for baseball ball.

Table 1. Confusion Matrix.

		Prediction outcome	
		P	N
Real value	P	59	1
	N	4	56

According to the matrix, on the left side, there is the user intention in the drawing and in the upper part, the value is given by the neural network. Note that the intersection of P with P represents the correctness of the model, whereas the intersection of P with N, its error. The same goes for N with P.

The matrix shows that the model hit the actual tests about 91.73% of the sample, only 3.27% below the tests with the dataset itself.

V. Conclusion

This work, which is a research in progress, has shown that using the classification results, it is evident how the use of classificatory models such as convolutional neural networks generates gains for the interface and human-computer interaction, making the user experience in virtual reality more immersive and funny. The real-time communication factor contributes to the user not losing the focus of the application, nor want to abandon it.

For future work, we want to add new categories of objects and allow the possibility of the game in online multiplayer format, to reinforce the usability of the drawing.

Acknowledgment

The authors' thanks to the Universidade do Estado do Amazonas for their support. The results were published through the research and development activities article of SAMSUNG OCEAN CENTER, sponsored by Samsung Electronics of Amazonia Ltda., with the support of SUFRAMA under the terms of Federal Law No. 8.248/91.

References

[1] G. Palhares Rodrigues and C. Porto, "Realidade virtual: conceitos, evolução, dispositivos e aplicações," in *Interfaces Científicas - Educação*, vol. 1, 06 2013.

[2] E. Bochinski, V. Eiselein, and T. Sikora, "Training a convolutional neural network for multi-class object detection using solely virtual world data," in *2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Aug 2016, pp. 278–285.

[3] X. Yin, P. Wonka, and A. Razdan, "Generating 3d building models from architectural drawings: A survey," vol. 29, no. 1. Los Alamitos, CA, USA: IEEE Computer Society Press, Jan. 2009, pp. 20–31. [Online]. Available: http://dx.doi.org/10.1109/MCG.2009.9

[4] C. Weichel, M. Lau, D. Kim, N. Villar, and H. W. Gellersen, "Mixfab: A mixed-reality environment for personal fabrication," in *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*, ser. CHI '14. New York, NY, USA: ACM, 2014, pp. 3855–3864. [Online]. Available: http://doi.acm.org/10.1145/2556288.2557090

[5] L. Y., B. L., O. G.B., and M. K.R., "Efficient backprop," vol. 1524, 1998.