Enhancement of Angry Birds Level Generation from Sketches Using Cycle-Consistent Adversarial Networks

Mury F. Dewantoro*, Febri Abdullah*, Pujana Paliyawan†, Ruck Thawonmas‡, Fitra A. Bachtiar§
*Graduate School of Information Science and Engineerin1, Ritsumeikan University, Shiga, Japan
†Research Organization of Science and Technology, Ritsumeikan University, Shiga, Japan
‡College of Information Science and Engineering, Ritsumeikan University, Shiga, Japan
§Faculty of Computer Science, Brawijaya University, Malang, Indonesia
ruck@is.ritsumei.ac.jp

***Abstract*—This paper presents our work to enhance a state-of-the-art level generator (Sketch-to-Level Generator) that generates levels for an Angry-Birds-like game from drawn sketches. To achieve this task, Cycle-Consistent Adversarial Networks (CycleGAN) are used. CycleGAN is trained using two datasets: sketch drawings and typical level-structures. The former are taken from Google's Quick, Draw! datasets, and the latter from the winning level generator at the 2017 and 2018 AIBIRDS level generation competitions. The output of the trained CycleGAN is used as the input of Sketch-to-Level Generator. Our results show that the proposed preprocessing technique using CycleGAN allows Sketch-to-Level Generator to more successfully generate levels from arbitrary sketch drawings.**

***Index Terms*—Angry birds, CycleGAN, sketch drawing**

I. Introduction

Many games nowadays have many interesting designs of their levels or stages, and many level-generation techniques have been implemented to make games more interesting. Here, we focus on a state-of-the-art example of such techniques that to generate a level from a drawn sketch (Stephenson et al [1]). In particular, their generator, henceforth called Sketch-to-Level Generator, transforms a human drawing to a level in a game called Science Birds, a clone version of Angry Birds widely employed for academic research [2]. Game levels generated from their generator will be similar to input drawings. Although Sketch-to-Level Generator can in most cases generate levels from input drawings that contain rectangle-shape objects, it struggles to generate a stable level when an input sketch contains objects with curve lines.

To solve the aforementioned issue in Sketch-to-Level Generator, this study proposes to first preprocess a given sketch using Cycle-Consistent Adversarial Networks (CycleGAN), developed by Zhu et al. [3]. CycleGAN is a powerful technique that learns a transformation between two image distributions. In particular, we expect CycleGAN to transfer a block-shape style, usually seen in Angry Birds levels, to the content of a given sketch, from which Sketch-to-Level Generator can readily generate a level.

II. Methodology

A. Quick, Draw!

Quick, Draw! (2016) [4] is an online drawing game by Google. This game uses neural networks to learn the player's doodling. In this game, the player is asked to draw a certain object, and the resulting drawing will be guessed by the neural networks. Currently, Google provides datasets of this game, containing 345 categories of drawings. Our study used parts of the Quick, Draw! datasets for training CycleGAN, in particular, those selected by extracting the first ten images in each category.

B. Level Generator

IratusAves is a level generator for Science Birds developed by Stephenson and Renz [5]. It was the winning entry at the the 2017 and 2018 AIBIRDS level generation competitions. This generator is capable of generating stable and solvable Science Birds levels with a high variety. In our study, to prepare a dataset containing typical Science Birds structures, IratusAves was used to generate 3500 Science Birds levels that contain no pig and TNT (explosive) objects, subject to a constraint that only rectangular block types are used.

C. CycleGAN

CycleGAN is a variant of Generative Adversarial Network (GAN). It is an artificial neural network that can perform image-to-image translation without pairing information, which facilitates the preparation of training data. Its developers publicly provide CycleGAN on both Pytorch and Torch implementations. This study applies the Pythorch version to train a model using Quick, Draw!'s sketch images and IratusAves's block-structure images as training data. During the training, given an image from the former dataset, the generator in CycleGAN is aimed at generating an image similar to a Science-Birds-like structure, and the discriminator attempts to gain the ability to detect that it is a faked one and label any of those in the latter dataset a real one.

D. Sketch-to-Level Generator

Sketch-to-Level Generator provides rectangular and non-rectangular modes for a given image input to generate a stable level. Their algorithm detects corners of the input sketch and identifies horizontal and vertical edges between detected corners. The identified edges will be reformed to a rectangle. Our study used the generator with the rectangular mode to validate results when CycleGAN is used and the non-rectangular mode otherwise.

III. Experiment

Our target sketch images are in black and white colors. As a result, we modified Science Birds blocks' colors to only black and white colors. In addition, we set Sketch-to-Level Generator to use only block types.

Due to the use of black and white images, we re-configured the input and output of CycleGAN to only one channel, rather than three channels for color images. We also examined the effects of applying a thresholding filter to the output of trained CycleGAN, where the pixels whose value is less than 220 will be black and the rest will be white; we empirically tried other values of the threshold, but the value of 220 gave us the most promising results.

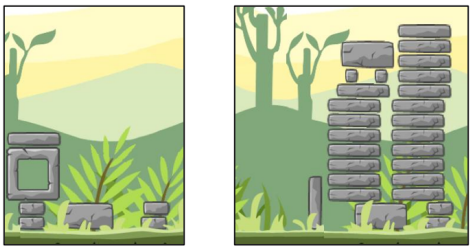
IV. Results

All levels were generated by Sketch-to-Level Generator. We evaluate our enhancement techniques by comparing levels between those generated by (1) a *baseline*, which is Sketch-to-Level Generator without CycleGAN, (2) Sketch-to-Level Generator with CycleGAN but without the use of the thresholding filter, and (3) Sketch-to-Level Generator with CycleGAN and the filter (CycleGAN+Filter). Our findings are in the following.

Figure 1 shows an example sketch image which is the input to either *baseline* or CycleGAN, the output from CycleGAN which is also the input to the filter, and the output from CycleGAN+Filter. Figure 2 shows a resulting level from each of the three methods. It can be obviously seen that structures from method 3 look more aesthetic and closer to the original sketch than those from method 2. In this example, a level cannot be generated by Sketch-to-Level Generator alone.

Table I compares the three methods in terms of the number of times Sketch-to-Level Generator can successfully generate levels with no error, the average number of blocks in a level, and the ratio of stable levels (those having no collapsing structures before a bird shot) over successfully generated ones. In total, 3450 sketch images were used. The two methods using CycleGAN, methods 2 and 3, could successfully generate levels much more than method 1. In addition, their levels have more blocks than levels generated by method 1. In terms of the total number of stable levels generated ($Level \times Stability$), method 3 has the highest number of stable levels.

Fig. 1: (a) an example sketch image, (b) CycleGAN's output, (c) CycleGAN+Filter's output



(a) **(b)**

Fig. 2: Science Birds structures generated from CycleGAN's output – **(a)** without the filter (cf. Fig. 1.b), **(b)** with the filter (cf. Fig. 1.c)

TABLE I: Comparisons between the three methods

	method 1	**method 2**	**method 3**
Levels	188	1762	2168
Blocks	4.54 $\pm$ 2.92	11.9 $\pm$ 10.78	10.15 $\pm$ 8.59
Stability	1.00	0.91	0.93

V. Conclusions and Future Work

From the results, the proposed system[1] with CycleGAN, regardless of the use of a thresholding filter, enhances Sketch-to-Level Generator. Nevertheless, resulting levels have certain possibilities to become unstable, thus unplayable. We leave this issue as our future work.

Acknowledgement: This research was supported in part by Grant-in-Aid for Scientific Research (C), Number 19K12291, Japan Society for the Promotion of Science, Japan.

References

[1] M. J. B. Stephenson, J. Renz, X. Ge and P. Zhang, "Generating Stable, Building Block Structures from Sketches," in IEEE Transactions on Games (p. 1).

[2] F. Lucas, Angry Birds clone (Science Birds). Github [Online]. Available: https://github.com/lucasnfe/science-birds [Accessed: January 30, 2020].

[3] J. Zhu, T. Park, P. Isola and A. A. Efros, "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks," 2017 IEEE International Conference on Computer Vision (ICCV), Venice, 2017, pp. 2242-2251.

[4] Google Creative Lab, "Quick, Draw!" Quick, Draw!. [Online]. Available: https://quickdraw.withgoogle.com/data [Accessed: January 30, 2020].

[5] M. Stephenson and J. Renz, "Generating varied, stable and solvable levels for angry birds style physics games," 2017 IEEE Conference on Computational Intelligence and Games (CIG), New York, NY, 2017, pp. 288-295

[1] https://tinyurl.com/abcyclecog2020