Towards the Development of AI Based Generative Design Tools and Applications

Juan Carlos Chacón(✉), Hisa Martínez Nimi, Bastian Kloss, and Ono Kenta

Graduate School of Engineering, Chiba University, Chiba, Japan
juancarlos@chiba-u.jp

Abstract. In recent years, several projects that take advantage of Artificial Intelligence as a design tool have arisen. However, most designers lack the technical knowledge necessary to profit from Artificial Intelligence in their design process fully. Through the development of GANSta, a tool with a graphical user interface that facilities the design and training of Generative Adversarial Networks. And the use and application of such a tool in different stages of the design process. By engaging in both iconographic branding element design and typographic font design projects. Participants of the Gesign lab initiative of Chiba University's System Planning Laboratory, explore the current and future opportunities that Generative Adversarial Networks present for their particular design process. Proving that previous knowledge in programming or machine learning is not necessary for designers to take advantage of the benefits that this technology presents from a generative design perspective.

Keywords: Generative adversarial networks · Generative design · Design tools

1 Introduction

Technology has always played a relevant role in the development of design, both at an academic and production level. In recent times, projects like Deepwear [1], which utilizes generative adversarial networks (GANs) to generate images of clothes that a designer can use as the basis for constructing clothing patterns, and the ChAIr Project [2], which uses a similar approach to generate chair shapes that designers can reinterpret and build, have taken advantage of artificial intelligence as a tool within the design process. Researchers such as Kevin German et al. [3] have also proved that AI, particularly GANs, can be used by designers in different stages of the design process.

However, most designers lack the technical skills or knowledge to fully understand the technology behind these projects, often perceiving it as magic [4]. These knowledge and skill gaps represent a relevant barrier that prevents designers, both professionals and those in training, from fully exploiting the potential of these types of AI tools. For that reason, Hughes states that developing design skills and knowledge of transforming technologies is integral for young people to not only understand but also design the future world [4].

E. I. Brooks et al. (Eds.): DLI 2020, LNICST 366, pp. 63–73, 2021.
https://doi.org/10.1007/978-3-030-78448-5_5

In addition, the lack of academic programs and activities that incorporate AI and machine learning technology into the design process has created a need for new learning approaches as well as the development of tools that facilitate the learning and application of AI-based approaches to design.

Thus, within the System Planning Laboratory of the Design Department of Chiba University, the Gesign Lab initiative emerged, involving a small team of academics and students who, through work sessions and small research project workshops, sought to design and develop methods and tools that enabled the use and learning of artificial intelligence concepts, particularly in relation to different design disciplines.

The following research outlines the process and results of the first stage of the project, which ended with the development of GAN Station (GANSta) [5], an internal tool that assists in the design and training of GANs in a graphical environment. This tool resulted from the process of developing two projects, the first focused on the design and generation of logo based graphical symbols through the use of generative adversarial networks, and a second focused on the design of a font character set based on the same approach.

2 Methodology

Project participants were divided into three groups based on their skills and interests: one team focused on tool design and development, and two teams focused on developing projects based on the resulting tools. The latter two teams selected different design outputs based on the personal interests of the participants, similar to a passion-based learning approach [6].

For the development of both projects, different GAN architectures, such as DCGAN [7] and CycleGAN [8], were first explored. After the preliminary results were obtained, StarGAN [9] was selected as the base for the three projects. This decision was made after training a network with a small dataset of 100 black and white images from two different domains, graphically evaluating the results of the interpolation between the domains, and visually and verbally evaluating the results as positive and interesting.

Once StarGAN was selected, a Python-based tool with a text command interface was developed to facilitate training and to test the models generated by the team members who lacked programming skills. This tool took a database folder with a set of images as an input and a simple training and test set of commands. With this first version of the tool available, the teams proceeded to develop the logo based graphical symbol project and the font character design project simultaneously.

2.1 GAN Based Logo Design Project

The objective of this project was to create a new brand logo based on Japanese family crests or Kamons for a Japanese tea shop. Team members proposed using GANs in order to help in the exploration and conceptualization processes and generated several proposals through a series of automated additions, alterations, and combinations of elements, mixing different domains. This approach allowed the team members to generate multiple proposals that, regardless of the selected domain or type of element, could be converted and synthesised to belong to the universe of the current Japanese family crest. During the exploration process, two different experiments were undertaken with different selections of datasets to assess the GAN's creative power. The first exploration process generated new graphic symbols mixing two domains so that the graphic symbol could include elements from two different categories.

The main objective of this first experiment was to generate a new graphic symbol of a teapot that included the characteristics of Japanese Kamons. The dataset was made of 200 black and white graphic symbols of 256×256 pixels that belonged to two different domains: Japanese Kamons and Teapots. As stated before, a StarGAN model was designed and used through the proposed command-based tool. This approach allowed the users to generate images from multiple domains and translate or apply a certain aspect or style from one image to another [10] (see Fig. 1).

In order to achieve this outcome, the team needed to train the generative model (G) to translate an input image x to an output image y conditioned by the target domain. In this first exploration, the total number of parameters for the generator was 8421120. For the discrimination, the total number of parameters was 44786624. The dataset was evaluated with Google Cloud's Vision API, which is called Vision AI [11]. Using this tool for image recognition, the team analysed how 100 images of Kamons and 100 images of teapots were labelled in comparison to the images generated by the proposed model.

The results of the Kamon dataset image analysis were the following: 67% of the images were labelled ‘Symbol’, 82% were labelled ‘Logo’, 29% were labelled ‘Emblem’, and only 5% were labelled ‘Crest’. In the case of the teapot dataset, none of the images were labelled ‘Symbol’, 26% of the images were labelled ‘Logo’, 98% were labelled ‘Teapot’, 67% were labelled ‘Illustration’, and 91% were labelled ‘Tableware’.

Finally, after training the discriminator and generator with the small dataset of 200 images, the trained model was tested. The evaluation of the results was performed by a designer through visual inspection, and a set of 100 images were selected to evaluate using Google's Vision API. During the image analysis, 9% of the images were labelled ‘Symbol’, 25% were labelled ‘Logo’, 0% were labelled ‘Emblem’, 69% were labelled ‘Illustration’, 37% were labelled ‘Teapot’, and 27% were labelled ‘Tableware’. The score of the results was lower on all the labels except for ‘Illustration’. Even though the second most recognized label was ‘Teapot’, none of the images generated by the proposed model were recognized as ‘Emblem’. However, despite the automated scores, the designer conducted a visual review and considered some of the results to be interesting visual proposals, concepts, and sketches for the creation of the logo.

For the second exploration, the main objective was to generate a new Kemon incorporating different pre-selected elements and to improve the quality of the images that resulted from the training process. For this goal, a dataset comprised of 640 black and white graphic symbols (256×256 pixels) belonging to 13 different domains was used [12]. The Japanese Kamons were divided into 6 different domain sets due to their similarities in shape and complexity. In addition, 7 domain sets were divided by their visual elements, such as teapots, cups, cakes, trees, leaves, and houses. Those domains were pre-selected by the team members due to their association with the brand personality

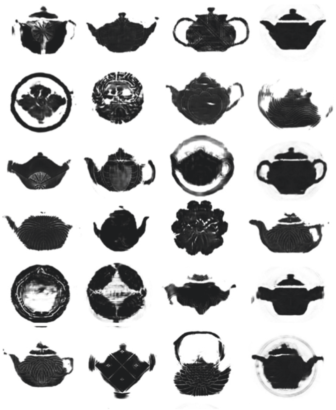

Fig. 1. Samples of graphic symbols resulted from the first exploration.

and brand values. The same StarGAN model based on the command line tool was also used, since it was possible to employ multi-domains while generating the images [14].

The GAN was trained using all of the symbols from the 13 domains, with the total number of parameters for the generator amounting to 8452480 and the total number of parameters for the discriminator amounting to 45114304. The evaluation process was done in three stages to analyse the obtained images. For the first stage, the results were evaluated by visual inspection, and 50 different samples were selected (see Fig. 2). Then, for the second stage of evaluation, the results were analysed by the Google Cloud's API, which is called Vision AI [11]. When compared to the first exploration, the second exploration showed a significant increase in label detection, with 32% of the images labelled 'Symbols', 56% labelled 'Logos', 8% labelled 'Emblems', and 82% labelled 'Illustrations'.

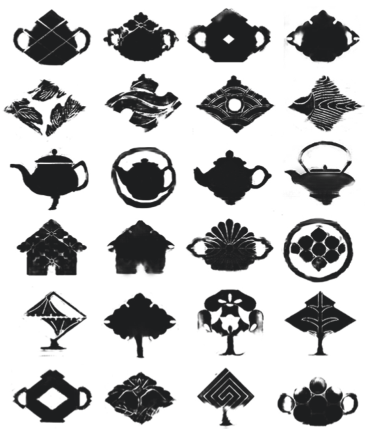

Fig. 2. Samples of graphic symbols resulted from the second exploration.

When compared to the results obtained during the first exploration, a significant increase in label detection can be observed. However, a third process was proposed for the evaluation of the results. For this last evaluation phase, a Kamon recognition model was generated to evaluate more accurately if the resulting images belonged to the Kamon category. In this model, 100 images from the Kamon dataset were annotated in the YOLO format and then used for the training of a custom YOLOv5 object recognition model with Kamon as a single category [13].

The resulting model, with a mAP@0.5 of 1.0, was then used to evaluate the 50 results previously selected. According to the results of the Kamon recognition model, 78% of the images were recognised as Kamons with an average confidence rate of 0.7533 over 1.0. This evaluation process contributed to validating whether the images generated by the proposed model could be considered to be symbols, logos, or emblems and specifically if they belonged to the category of Japanese family crests. However, a qualitative evaluation of graphic symbols is also an essential and determining factor in graphic identity design.

In conclusion, the selection of dataset images, the number of domains, and the quality of those images have a direct impact on the quality of the results and are important factors to take into consideration for possible future work. This approach showed that machine learning (and specifically GANs) can be a tool for the branding design process and that graphic design processes could benefit from using GANs to automate and improve their results.

2.2 GAN Based Font Design Project

For the second project in the program, participants envisioned a tool capable of creating a character set with one character as the input. This tool could help a type designer to improve the font creation process by generating a whole character set after only the first character is designed. Similar to the logo project, a Stargate that takes images and manipulates them to fit certain characteristics of different domains, including emotional features on the input image, was designed and used [9, 10, 14].

Using different domains made it possible to create a dimension for every character in the set. Thus, it was possible to take the characteristics of the input and adapt the shape of its different dimensions. To prepare the dataset, the team collected a set of font files. For this project, a master folder of Google fonts was used [15]. The font files were divided into different folders by their category and style.

Aiming for a legible and usable font as the final output of the project, the team decided to only use sans serif fonts in regular style as part of the dataset in order to keep it as simple as possible. All of the glyphs in the font dataset were extracted as PNGs with a size of 256 $\times$ 256 pixels [16]. The dataset was then loaded into the GAN, and the training started. After 1,000,000 training iterations, the characters could be identified in every line produced in a resulting image generated by the GAN. A total of 5,996,000 iterations were trained. Training took 26 days, 10 h, and 27 min until the team decided to interrupt the process for a visual evaluation of the results. Figure 3 shows the results of the 5,400th module.

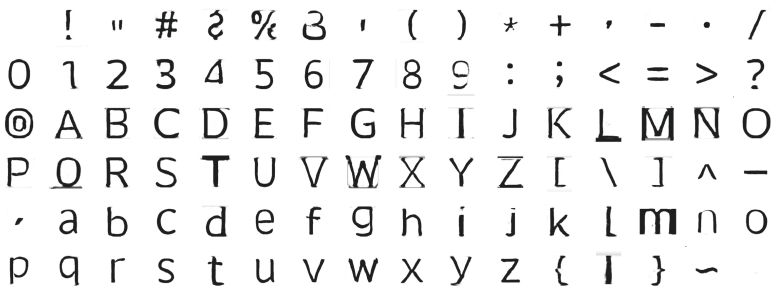

Fig. 3. Character samples of the 5,400th training module.

The input that was used to generate these characters was a basic round dot. Figure 4 shows the outputs from the same module with different input variations. The first one shows the result of using a very basic shape, in this case a dot, and the second sheet shows the characters that were generated with a minor äún, äù as the input.

Fig. 4. First output samples generated by the model.

One important finding from this project was that using simpler shapes as the input generated visibly cleaner elements and using more complex shapes caused glitches that resulted in less recognizable characters. For the evaluation, the team created 12 characters of differing complexity to test. The characters were three different versions of a dot, a number one, an exclamation mark, and a minor n. After visual inspection, the team could visually identify that the complexity of the input had a direct influence on the quality of the generated results. The dots, the simplest elements used, had the highest quality outputs.

As the final result, a set of 96 characters corresponding to the basic Roman numerals and letters were generated as a set of PNG files that were then vectorised as a font file, as shown in Fig. 5. Several noise adjustments and flaws, particularly in the baseline and kerning, remained in the final font file as a result of the image generation and vectorising processes. However, as a proof of concept, the team considered the project to be a successful exploration.

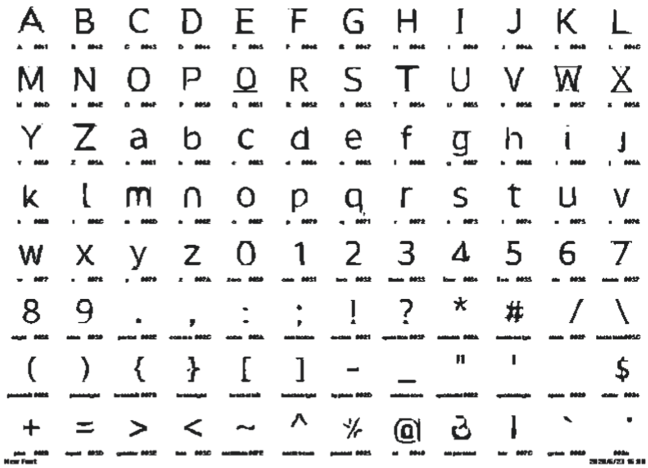

Fig. 5. Final 96 characters generated by the model.

2.3 GAN Station Project

Once the command-based tool was established as the base for the iconographic and typographic projects, the training and evaluation processes for the generated models needed to be improved. To this end, using the code and instructions originally implemented in the command-based tool as a base, a minimum viable prototype of a tool with a graphical interface that would make the process easier and more accessible was developed.

Thus, the GAN Station project, GANSta, emerged, a name voted upon by the participating team members at the time. A node-based interface was adopted due to the familiarity of the team members with this type of interface in tools such as TouchDesigner [17] and Grasshopper [18], as well as for the ease with which these interfaces presented their programming logic, specifically through the use of blocks, which could facilitate the learning of the concepts behind the development of a GAN for future users or participants who lacked prior knowledge on the subject (see Fig. 6).

In the development of GANSta, Anaconda [19] was used with Python 3.7 as the environment, and PyQt [20] was used as the framework for interface building. GANSta's final design consisted of a user interface with the main window divided into a work area and a side menu (see Fig. 7).

Within the work area, users can place, drag, and drop the nodes necessary for the construction of their GAN from the left side window. These nodes consist of the following:

- Dataset node. Its function is to select and verify the path in which the dataset to be used for training is located. This dataset consists of a series of images divided into two folders: one named 'Training' for the images to be used during the training process and one named 'Test' for the images to be used during the test process. Inside the training folder, the folders with the names for each category or domain to be considered must contain their respective images. This node output connects to the Image node.

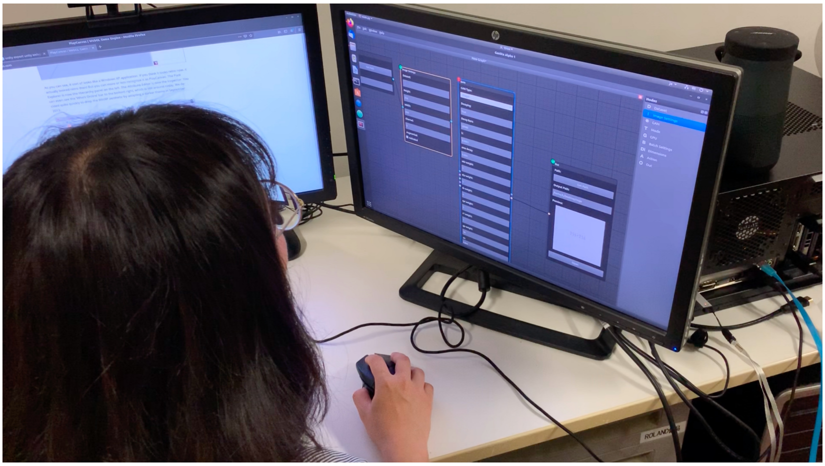

Fig. 6. Student using the GANstation tool at System Planning Lab. Chiba University, Japan.

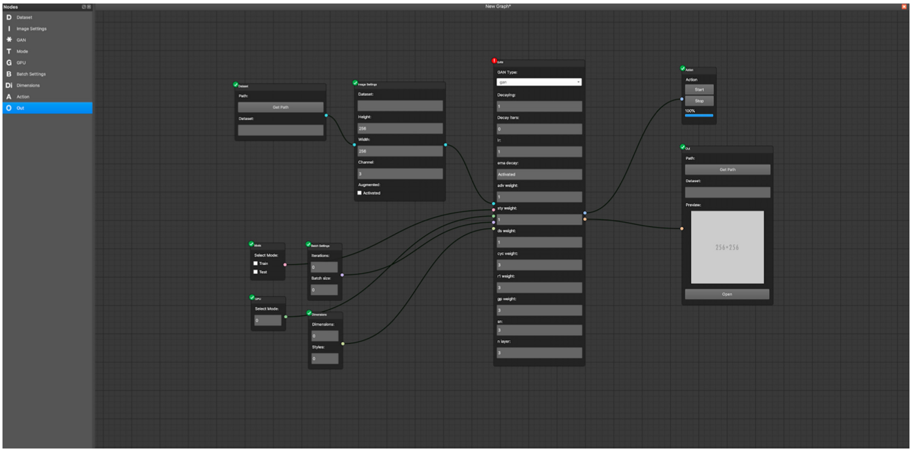

Fig. 7. GANstation user interface.

- Image node. Its function is to define the parameters, such as width and height, of the types of images to be used during the training process and to allow the user to increase those parameters during training so as to distort or randomly rotate the images to avoid problems derived from the lack of variety in the dataset. This node output connects with the GAN node.
- Mode node. Its function is to allow users to select whether they want to train the model or test its results. This node output connects with the GAN node.
- GPU node. This node is optional. Its function is to allow users to select whether or not they want to use GPUs for the training process. It also shows the number of GPUs available on the user's machine. This node output connects with the GAN node.

- Batch node. Its function is to determine the number of batches and epochs needed for the model training. This node output connects with the GAN node.
- GAN node. Its function is to determine the parameters of both the generator and the discriminator, as well as to verify that all of the parameters of the previous nodes are valid or correct. This node output connects with the Output and Action nodes.
- Output node. Its function is to define the path and format of the training results, as well as to provide an internal window to show a preview of the results of the training or testing processes. This node does not connect to the other nodes.
- Action node. Its function is to start the training and testing of the model based on the option selected by the user, as well as to show the percentage of the process completed. This node does not connect to the other nodes.

To use the tool, the user must prepare their respective image-based dataset, as indicated in the dataset node description, and then design its GAN by dragging and dropping the desired nodes into the action window and modifying the parameters of the node as desired. Once the first version of the graphical tool was developed, it was tried and tested by project team members using their existing approach and generated datasets, in order to evaluate the necessary changes in the interface to facilitate its use. As a result of the requests made by the project participants, some changes were made, resulting in a second version of the graphical tool. Among these changes, the Output node was removed, and the functions of both the Output and Action nodes were transferred to a second side menu.

3 Discussions and Conclusion

Although there are other graphical-based tools for programming and interacting with artificial intelligence models [21, 22], the main difference between GANSta and other tools for the development of machine learning models radiates in the fact that GANSta was developed based on the necessities identified during the development of two design-led projects as an internal tool for designers. Seeking to facilitate not only the understanding of the concepts behind the development and operation of the GAN but also facilitating the learning experience, by allowing designers without previous knowledge of programming or machine learning to develop and train their own GAN based models through interactive exploration. Encouraging incorporating the designed tool within their design process.

By designing new artificial intelligence tools to generate iconographic elements and typographic characters for logos and by providing a prepared environment with the necessary instructions to design and train generative adversarial networks (GANs), the participants in the Gesign Lab initiative have proved that both professional designers and design students without prior knowledge of programming or artificial intelligence can identify uses and applications for these tools both in the design process and in the execution of final pieces. However, there are still many limitations beyond the technical requirements of this type of AI tool that may hinder its widespread adoption, such as the cost of the computer equipment, the number of graphic processing units required to execute the training process of the GANs, and the time that the training process takes.

This lengthy process may make a tool such as the one developed for this study unattractive for activities that go beyond experiments or explorations of how design processes could change in the near future. Nevertheless, the fact that designers such as the participants in the Gesign Lab initiative could successfully develop this type of tool shows that it is only a matter of time until artificial intelligence is employed more widely in the design process.

References

1. Kato, N., Muramatsu, N., Osone, H., Ochiai, Y., Sato, D.: DeepWear: a case study of collaborative design between human and artificial intelligence. In: Proceedings of the 12th International Conference on Tangible, Embedded, and Embodied Interaction vol. 2018, January, pp. 529–536 (2018)
2. Schmitt, P., Weiß, S.; The Chair project: a case-study for using generative machin learning as automatism. In: 32nd Conference on Neural Information Processing Systems (NIPS) (2018)
3. German, K., Limm, M., Wölfel, M., Helmerdig, S.: Towards artificial intelligence serving as an inspiring co-creation partner. EAI Endorsed Trans. Creat. Technol. **6**, 162609 (2019)
4. Hughes, J., Robb, J., Lam, M.: Making future-ready students with design and the internet of things. EAI Endorsed Trans. Creat. Technol. **6**, 163096 (2020)
5. Chacón, J.: GAN Station. A node based generative adversarial network tool. (Version 0.1). Zenodo (2020). https://doi.org/10.5281/zenodo.3929374
6. Brown, J.S., Adler, R.P.: Minds on fire: open education, the long tail, and learning 2.0. Educ. Rev. **43**(1), 16–32 (2008)
7. Metz, A.R.L., Chintala, S.: Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. ICLR (2016)
8. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision, vol. 201, pp. 2242–2251 (2017)
9. Choi, Y., Choi, M., Kim, M., Ha, J.-W., Kim, S., Choo, J.: StarGAN: unified generative adversarial networks for multi-domain image-to-image translation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8789–8797 (2018)
10. Banerjee, A., Kollias, D.: Emotion Generation and Recognition: A StarGAN Approach, Imperial College London (2019)
11. Google Vision API. https://cloud.google.com/vision. Accessed 23 June 2020
12. Chacon, J., Martinez Nimi, H.: Kamon Dataset and StarGAN Model Version 1.0 Data Set, Zenodo (2020)
13. Chacon, J., Martinez Nimi, H.: yolokamon v1.0 A Yolov5 Japanese Kamon detection model (2020)
14. Choi, Y., Uh, Y., Yoo, J., Ha, J.-W.: StarGAN v2: Diverse Image Synthesis for Multiple Domains (2019)
15. The Google Fonts Team: Google Fonts Files. https://github.com/google/fonts. Accessed 23 June 2020
16. Chacon, J, Kloss, B.: San Serif Font Dataset and StarGAN Model Version 1.0 Data Set, Zenodo (2020)
17. TouchDesigner [Computer software]. https://derivative.ca (2020)
18. Grasshopper [Computer software]. http://www.grasshopper3d.com (2020)
19. Anaconda Software Distribution: Computer software. (Vers. 2–2.4.0). https://anaconda.com (2016)

20. PyQT: Cross-platform software development for embedded & desktop. https://www.qt.io/. Accessed 17July 2020
21. RunwayML [Computer software]. https://runwayml.com (2020)
22. ML5 [Computer software]. https://ml5js.org (2020)