Image Co-Creation by Non-Programmers and Generative Adversarial Networks

Doron Friedman[a], Dan Pollak[a]

[a]*The Advanced Reality Lab, The Interdisciplinary Center, Herzliya, Israel*

Abstract

Generative models such as generative adversarial networks are now being studied extensively. Eventually, however, many of them are intended for non-programmers to work with, e.g. designers, artists, or other content creators. What happens when such individuals are confronted with using GANs? We present a case study – a new course intended for non-programmer MA students in human-computer interaction, aimed at training them in authoring content using generative models. As their final assignment, the students were asked to train generative adversarial networks in order to generate images from a predefined category of their choice. The students either used a graphical user interface (GUI)-based software or modified preexisting python code using simplified Google Colab notebooks. We present several lessons learned from this course. First, we analyze the joint human-AI creation process and recognize points where students could intervene, with anecdotal examples of how they creatively explored these opportunities. Interestingly, while the majority of algorithmic research is focused on how to make models more controllable (e.g., via conditioning or latent space disentanglement), the students found ways to obtain their creative needs by mostly exploring the dataset level (as opposed to the model architecture). Additionally, we present the results of a short survey, comparing the two modes of work (GUI vs code).

Keywords

GAN, co-creation, Style-GAN

1. Introduction

We are witnessing a rapid advance of “artificial intelligence” (AI) and machine learning (ML) technologies, and these techniques are penetrating into an increasing range and diversity of aspects of everyday life. It seems important that the responsibility for these systems would not only lie on the shoulders of programmers, but that additional professions would be involved in intelligent system design, development, and evaluation. As the current zeitgeist is that of data-driven methods with “deep” neural networks, explainability has become a major concern [1]. In generative AI, rather than just explainbility (or instead?), we often strive for ‘controllability’, or the degree to which humans can control and shape the results generated by the system. Indeed, there is ample work on on reversible generative models [2] or latent space disentanglement [3]. Nevertheless, in addition to these computational efforts, sociological factors are expected to play an important part. Eventually, these systems are not intended for programmers; rather, they would more likely be used by designers, artists, writers, or other professionals of the so-called ‘creative industries’.

Our goal is to explore how such individuals can work with novel generative models such as generative adversarial networks (GANs). The opportunity came up in the form of a course for MA students in human computer interaction (HCI), most of them non programmers. As their final assignment in the course they were guided in developing a project in which they train a GAN to generate a specific set of images, in the spirit of “This X does not exist”[1]. Following the success of StyleGAN to generate highly photo-realistic human faces, a website showing such images called “thispersondoesnotexist” went viral. This was followed by attempts in training GANs specializing in generating cats, rental apartments, snack, and the list of project seems to be still growing. The task is similar to the Little Prince’s request: “draw me a sheep”. Unlike the majority of human-AI co-creation tasks, in which the human is expected to be creative and the machine is expected to assist, in this task the machine is expected to be creative and generate interesting samples of X, and the human is only expected to assist. The students were given the choice whether to use programming or a GUI-based software, and following the course, they were asked to answer some questions.

This paper’s contribution is from lessons learned from teaching generative models and synthetic media to non-programmers, anecdotal lessons learned from their projects, an analysis of the human intervention points that they ‘discovered’, and results from a sur-

Joint Proceedings of the ACM IUI 2021 Workshops
doronf@idc.ac.il (D. Friedman); dandan888@gmail.com (D. Pollak)

[1] http://thisxdoesnotexist.com

vey in which they provided some feedback[2]. Interestingly, while the overwhelming majority of algorithmic research is focused on how to make models more controllable (e.g., via conditioning or latent space disentanglement), the students found ways to obtain their creative needs by mostly exploring the dataset level (as opposed to the model architecture).

2. Background

The "draw me a sheep" scenario raises several questions. The first is: can machines be creative and if so what does it mean? Generative algorithms are able to generate a huge number of outcomes; however, it could be that most or even all of the results are uninteresting. Boden [4] stressed that in order for a process to be considered creative its result needs to be both useful and novel. Satisfying the requirement for novelty is typically easy (e.g., throw in some noise, or increase the exploration temperature). Whether the result is useful is hard to define, in general. Additionally, GANs are (arguably) not designed to be creative; on the contrary, they are designed to learn the training set prior distribution and come up with the most prototypical and 'non-creative' examples.

Thus, we suggest that machine creativity is not the most appropriate framework for our case study, but rather the appropriate framework is human-AI collaboration. Here we can distinguish between two approaches. The first approach is more human-centric, where AI is expected to augment human creativity. For example, Jarrahi [5] discusses the AI job crisis in general, stressing the need for a discussion on how humans and AI can work together, and the role of AI in human intelligence augmentation rather than replacement. However, Jarrahi also presents examples of human-machine symbiosis in chess and medical diagnosis, and discusses the synergy in organizational decision making.

This leads us to the second, more recent approach, where there is growing interest in human-AI collaboration as a new entity, where the whole is larger than the sum of the parts. Bidgoli et al. [6] draw historical lessons from collaborative art, where groups of artists created new identities that attracted significant attention in the 1960s and the 1970s. This historical analogy can shed light on contemporary human-AI projects. For example, cultural diversity was highly encouraged in those artist groups; clearly, humans and machines are inherently very different from each other, which may lead to superior results. We suggest, however, that there are also important differences. For example, in those artist groups the importance of the individuals comprising the group was much reduced; however, in our case we are still interested in distinguishing the role and contribution of the machine and the human. Bidgoli et al. actually suggest that the machines (at least in this point in time) do not have an identity to contribute; rather, it is the identity of the people who designed it. "From this point of view, co-creation is a special case of collaboration where the tool acts as a "machinic surrogate" to represent the identity of its "toolmakers"." We suggest that this, again, undermines the important differences between humans and machines.

Interestingly, the current legal perspective seems to agree that the results of human-AI collaboration may result in emergent properties that cannot be attributed to any of the stakeholders; Eshraghian [7] looks at ownership of AI-art from a legal point of view, pointing out that the stakeholders include programmers, style trainers, private datasets, and features extracted from information collected from the general public.

Most often, there is a sociological gap between algorithm researchers and the target audience for using these tools, which are, in general, non programmers. We anticipate that as 'generative AI' will play a larger role in an increasing number of domains, this gap will need to be bridged, and ideally it should be addressed by the research community, not only by industry teams working on specific products. In the domain of music generation, Huang et al. [8] report on a survey carried out with musician/developer teams, and the same team evaluated a tool they developed for co-creation in song writing [9].

The human-AI co-creation model has drawn increasing interest recently [10]. The main requirement is that the outcome cannot be attributed to either human or machine alone; we suggest that our case study adheres to this requirement, as illustrated in the results section. Otherwise, however, our case study is very different from other research projects in the field of human-AI co-creation, and we suggest that the assumptions and range of questions that addressed by this field may be extended. In other words, we suggest that the field might be too 'human-centric', in both the end goal (the focus is on tools that enhance human activities) and the means (the focus is on real time interaction techniques).

For example, Karimi et al. [11] suggest a framework for evaluating human-AI collaborative creativity. Their taxonomy suggests three types of systems: fully autonomous, creativity support tools, and co-creative systems. However, their definition of co-creativity re-

[2]See video: https://bit.ly/3qe2mf5

quires the AIs to have 'their own conceptualization of creativity', which we suggest is not a necessary component. Most work in the field is based on tools whereby humans and AI interact directly, in real time: Yannakakis et al. [12] demonstrated joint exploration of the search space, and other studies also present interactive tools for painting or sketching [10, 13]. However, they typically go too far (we suggest) in requiring the AI to have explicit mental models and natural language communication abilities. We suggest that this is at most desired, but there are more fundamental questions in human-AI co-creation. Llano et al. [14] suggest that creativity can be enhanced by improved communication between human and machine; the end goal is two-way communication. We suggest that at this stage this is at most a probable hypothesis.

Co-creativity has also been framed as mixed initiative interaction [12, 15]. The mixed initiative framework, in turn, has been borrowed from conversation analysis, and includes three parts: task initiative – deciding the topic, speaker initiative – deciding who speaks, and outcome initiative – deciding when the outcome is ready [16]. In our case, all three decisions are exclusively made by the human (although, see the results section for a caveat). Nevertheless, we suggest that the "draw me a sheep" scenario still falls under the original requirement – the whole is more than the sum of the parts, and it is difficult to disentangle the relative contribution of human and machine when analyzing the result.

Instead, we suggest that an appropriate framework is to view the human-AI collaboration as a single process. The analysis we perform (in Section Results) is aimed at identifying the sub tasks in this creative process, and who is in charge of each sub task. In our case the machine would ideally be autonomous, but we see that there are quite a few points at which human intervention is desired or even required.

3. Method

3.1. The Students

The course was an elective as part of an MA degree on human-technology relationship. Thirty two students enrolled in the course, most of them with a background in design or social science, and only a minority with programming experience or computer science background. All students learned basic programming (with p5js[3]) in the first semester, and the course was

[3]http://p5js.org

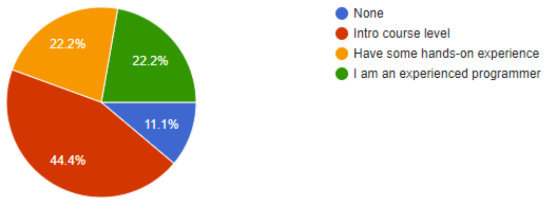


Figure 1: Programming experience of 18 students who filled in survey questionnaire.

given in the third (summer) semester. Figure 1 describes their level of programming experience. All students provided written consent for their material to appear in writing and video.

3.2. The Course

The course was focused on "synthetic media", i.e., automatically generated media and art content[4]. The first two lessons provided a historical view of AI, introducing key concepts and themes. The third lesson provided an overview of AI used for generating media and art content; while the course focused on deep neural networks, in this stage we also let the students explore genetic algorithms hands-on, allowing us to discuss generative projects such as Karl Sims' 3D evolving creatures [17], and explaining that neural networks are one technique among many. Next, we provided a brief introduction to ''classic ML' and deep neural networks, and the second half of the course was dedicated to a more in depth discussion of GANs and some additional topics ("deep fake", sequences, language models). We have introduced some mathematical notation, for example, discussing loss functions, but the course was intended for non-mathematicians and most of the discussion was at the level of popular science. Due to the Covid-19 pandemic most of the lessons were hybrid, with half of the students in class and the rest at home over Zoom.

In their assignments, the students explored both an easy-to-use software – RunwayML[5], and a simple Colab notebook with python code. RunwayML is a commercial product intended for artists and creators to apply a wide range of "deep learning" models. It provides a relatively easy to use graphical user interface (GUI) that serves as a front end to existing implementations of deep neural networks. It allows you to run the trained models on your input, to train models on new datasets, and to concatenate models; i.e., you can

[4]See video: https://bit.ly/3qe2mf5
[5]http://runwayml.com

concatenate networks A and B if the type of output of network A is consistent with the type of input of network B. The payment is mostly per cloud GPU runtime. Such services raise interesting and important questions around copyright and ownership of intellectual property, which are out of the scope of this paper.

As a first generative assignment the students were asked to explore "deep dream" [18], using both Runway ML and a Colab python notebook. In both cases the level of exploration was very minimal – the students could only select the image to modify and the Inception layers whose activation would be maximized. For the final assignment the "brief" was to come up with images in the spirit of thisXdoesnotexist[6]. The students were provided with a Colab python notebook for scraping images online. Next, they were told to choose between two options: a notebook with documented deep convolutional GAN (DC-GAN) [19] implementation, and RunwayML. It was explained that the code allows for more freedom, whereas using StyleGAN [20] or StyleGAN-2 [21] on RunwayML would produce better results. Nevertheless, we explained that RunwayML is also constrained by cost and we warned the students that the extra cost we could cover per team is limited. DC-GAN is close enough to a 'vanilla' GAN so that there are simple implementations that can be documented and explained. The implementation of StyleGAN, on the other hand, is only accessible to experienced programmers (wrappers are available as simple notebooks that allow you to run the model or even train it, but this is functionally almost equivalent to using RunwayML, and does not serve any educational purpose of understanding neural network coding).

The programmers were assigned to separate teams, and they were not allowed to use RunwayML. They were told to explore a range of techniques and hyperparameters and provide documentation of training runs, such as plots of generator and discriminator loss. Most of them started with DC-GAN but moved on to StyleGAN-2 in order to improve their results.

3.3. Questionnaire

A week after the submission of the final project, the students were asked to fill in a questionnaire, and 18 students responded. We asked about programming experience and what option they used for their final project. Next we asked them to rate, for either python code or RunwayML, the extent to which: i) they liked using it, ii) they found it difficult, iii) the results matched their expectation in quality, and iii) the results matched their expectation in intent.

[6]http://thisxdoesnotexist.com

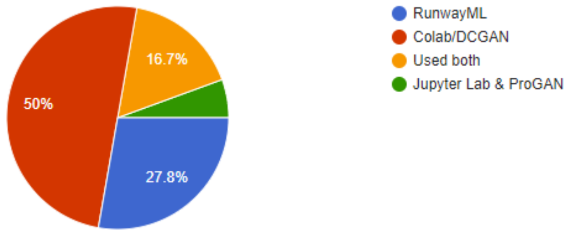


Figure 2: The software used by teams for main assignment.

4. Results

Figure 2 describes whether the students used RunwayML, python, or both. Interestingly, some non-programmers (i.e., students whose only very basic programming experience was in p5js, from a previous course) preferred to use Colab.

4.1. Questionnaire Results

In order to find out the differences in responses to using RunwayML versus using python and Colab we ran paired-samples t-tests. The students liked RunwayML significantly more than using code (t=3.073, df=17, p=0.007). Perceived quality was significantly higher with RunwayML (t=2.309, df = 8, p=0.05) and RunwayML was significantly easier to use (t=-3.274, df=7, p=0.014). However, there was no significant difference in the extent to which the user's intent was captured by the model, when comparing the two platforms (t=1.0, df = 8, p=0.347). Nevertheless, the correlation between perceived quality and captured intent was high and significant for both RunwayML (r=0.76, df=12, p=0.003) and Colab (r=0.66, df=13, p=0.01).

4.2. Lessons Learned from Student Projects

We analyze the projects in terms of the points where the human users could intervene, with focus on the non-programmers (including students with some very basic programming experience). The first human intervention point was in determining the goal – allegedly, this was completely determined by the humans. However, it could be argued that the machine 'took part' in this stage as well, since about half of the teams modified their project goal after some attempts. Typically, after realizing that their initial expectations were not realistic, the students converged on more specific outcomes, since they realized that the datasets have to be homogeneous for obtaining good results. Eventually, the non-programmer teams decided to generate: human head statues (protomes), holocaust victims, Gaudi-style architecture, animal-electronic device hybrids, city maps, Simpson characters, politicians, butterflies, protected flowers, Disney characters, smartphone application icons, and bicycles. The programmer teams decided to generate: Marvel superheros, Yayoi Kusama art, best of art paintings (unspecified), and McDonald's toys (mixed team).

The students quickly learned that the main way for them to affect the results is by modifying the training set. Although this was discussed in class, many were surprised that the datasets need to be very homogeneous and that "the AI" could not deal with simple invariants such as location of the main object within the frame. Some surprising artifacts were discovered; for example, the Gaudi team had to clean cranes from pictures of the Sagrada Familia, which has been under construction for a long time. Sometimes such biases were considered undesired, but in other cases the students were excited to see these biases emerge from the model's results. For example, it was considered a success to witness that the GAN had incorporated flags into many images of generated politicians, and often placed them in a 'speech stance', thus capturing stereotypical features of politician pictures that distinguish them from other pictures of humans (see Figure 3).

The next choices were technical: what algorithm to use (the students were pointed to DC-GAN, Style-GAN, and StyleGAN-2), and automatic image preprocessing, which still left decisions regarding image size and crop type.

For Style-GAN based projects the networks were always pre-trained, and an important decision was what pre-trained model to use. Sometimes the decision was obvious, but in some cases students explored what happened when they override the most reasonable decision. For example, the Disney team wanted to generate whole body images and avoid photo-realistic faces, so they opted for a model pre-trained on animals rather than realistic human faces (Figure 4). As another example, the Gaudi architecture team realized that using a model pre-trained on skyscrapers resulted in no green (trees and plants), so they preferred the results obtained with a model pre-trained on objects (Figure 5). The butterfly team realized that it is relatively easy to generate beautiful butterflies, but using a model pre-trained on faces resulted in more symmetric butterflies (Figure 6a) than when using a model pre-trained on objects (Figure 6b).

Figure 3: Generated politician images reveal politiican image distribution properties: typical upper body composition, formal dress, US flags. Results obtained with StyleGAN-2 in RunwayML, by Nurit Belorai.

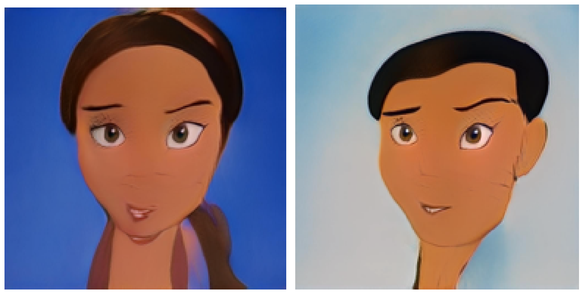
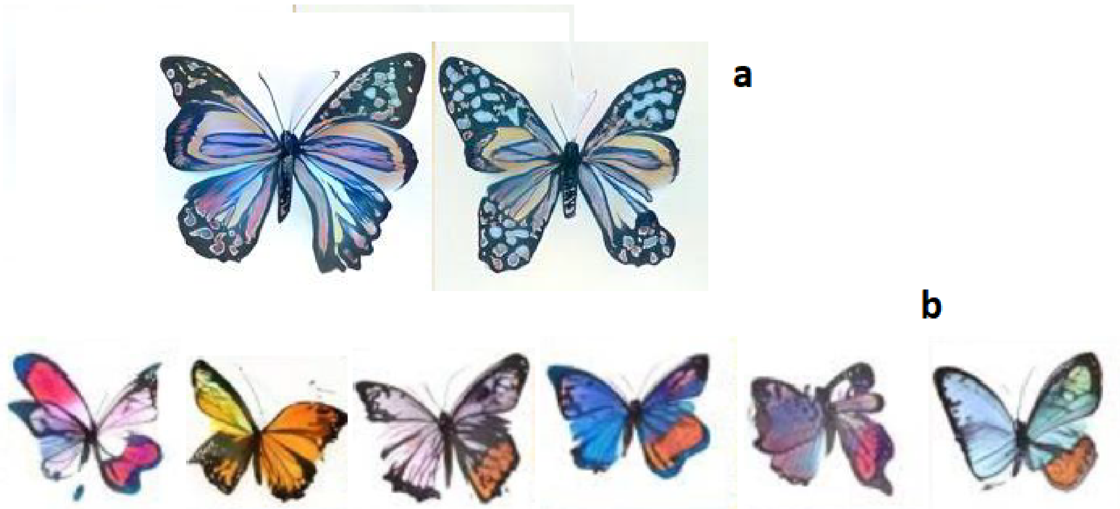


Figure 4: Generated Disney characters; trying to generate characters with bodies using a model pre-trained on animals failed in producing clean results. Results obtained with StyleGAN-2 in RunwayML, by Hadas David and Shani Tal.

One team deliberately trained a model pre-trained on one category (cats or horses) with a very different dataset (toasters or kitchen aid devices), with the aim of creating a hybrid animal and electronic device (e.g., toaster cats in Figure 7). They realized that training over many steps resulted in interpolating the model from one category to another, and they have specifically tried to find the point in time (number of training steps) where the balance is of interest, systematically testing when the horses disappear into kitchen

a b

Figure 5: Generated Gaudi-style buildings. Training on a model pre-trained on objects preserved green (trees) (a) in the images (a), while training on a model pre-trained on skyscrappers (b) did not. Results obtained with StyleGAN-2 in RunwayML, by Nachshon Ben and Carmel Slavin Brand.



a

b

Figure 6: Butterflies generated by a model pre-trained on faces were symmetric (a) while butterflies trained on objects were often not symmetric in shape and color (b). Results obtained with StyleGAN-2 in RunwayML, by Gizem Odemir and Karin Bar-Gefen.

aid (they were happy with the results obtained after 680 steps; Figure 8). While there has been computational efforts to train GANs on such mixed datasets, the goal in those cases was to teach the GANs to learn separate modes [22]; the question of deliberate "mode mixing" seems novel.

Dataset level user manipulations are not popular in the field; one of the main reasons is most likely that re-training is resource (time) consuming. Each 'experiment', even if it only involves fine tuning, typically takes at least several hours. Clearly, this has an effect on the nature of the creative process, making it essentially 'non-interactive' and limiting the scope of the work to a small number of iterations.

The next choice was the number of training steps or epochs. The students realized that more training is not necessarily better; while most often the convergence is to a flat curve (i.e., the model keeps generating similar images regardless of additional training), sometimes earlier results are preferred. The only information they could use in RunwayML, other than manually inspecting the resulting images, are the Frechet Incep-

Figure 7: Toaster cats, an example of attempts to create hybrid animal and electronic device images. Results obtained with StyleGAN-2 in RunwayML, by Eden Offer and Adi Frug.

Figure 8: An example of attempts to create hybrid animal and electronic device images – horses and kitchen aid. Results obtained with StyleGAN-2 in RunwayML, by Eden Offer and Adi Frug.

tion distance (FID) scores (only programmers learned to plot and analyze additional information, such as learning curves).

One team realized that they can keep fine tuning the network more than once, over multiple steps. First, they generated butterflies form models pre-trained on faces or objects. Next, they wanted to further shape the results, with the aim at generating butterflies with fractal patterns (failed) or animal patterns. They continued training their fine-tuned butterfly model for a smaller number of steps with pictures of animal skin. Moreover, they realized that they could easily cut those pattern pictures into a rough shape of a butterfly, so that the results will not lose the butterfly shape (Figure 9).

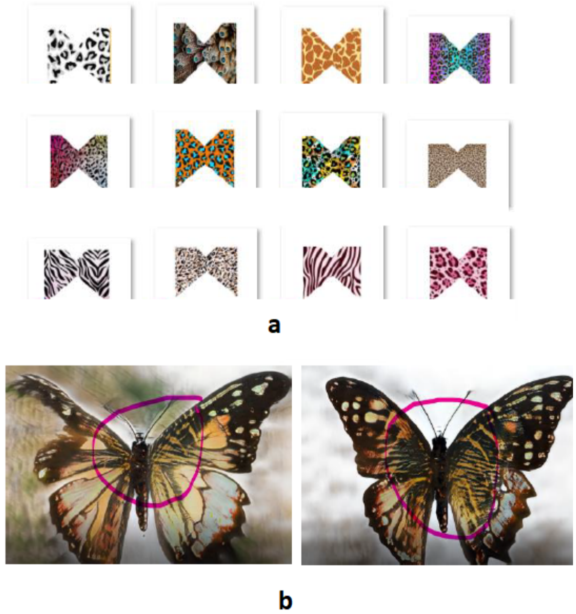


Figure 9: Fine tuning butterflies with different patterns: patterns from training set (a) and resulting butterflies with tiger stripes (b). Results obtained with StyleGAN-2 in RunwayML, by Gizem Odemir and Karin Bar-Gefen.

After the training process, the students learned the controversial art of 'cherry picking'. While in most academic contexts this practice should be avoided, we

Figure 10: Automatically generated applications icons: the results were low quality (left) so they were recreated manually (right). Results obtained with DC-GAN in Colab, by Maor Bluman and Bat Primo.

suggest that in the context of AI-aided design or art this is quite legitimate. It is important to understand the implications of the systems being generative, i.e., once developed they can generate a practically infinite number of results. If one of the results is what you are looking for, and what you are looking for is very special and difficult to obtain, "cherry picking" is reasonable. Finally, one team of non-programmers who opted to use DC-GAN were rightly disappointed from the results, which were noisy. Nevertheless, they went on to manually clean the results, suggesting that even if the AI 'artist' or 'designer' is not as competent as the human, it can nevertheless be used as a source of inspiration (Figure 10).

5. Discussion

Non-programmers were able to grasp the main concepts and train GANs to obtain interesting results, using either a GUI-based software intended for non-programmers or by slightly modifying pre-exisiting simple code. Questionnaire results indicate that using the GUI with StyleGAN-2 was easier than code and resulted in better perceived quality. Interestingly, there was no significant difference in the perceived degree to which the results matched the original students' intent. This is not because intent was very low, because the mean reported intent is higher than the average, and perceived quality and intent are highly correlated. Nevertheless, our assessment in this respect is limited, because we are not only comparing two tools, but in many cases we are also comparing two models – most students who opted to use code used DC-GAN, whose results are inferior as compared to StyleGAN. We suggest that our course is not only useful for academic institutions but may also be useful in industry, for training non-programmers to co-create with AI.

We have analyzed the human intervention points in the creative process. The students were required to intervene in several sub-tasks that were not implemented in software. Additionally, some students intervened in order to refine the results. While we perceived the task in the context of almost autonomous AI, at least two teams interpreted the task in terms of AI assisting human creativity: one project aimed at augmenting the design of electronic devices with inspiration from animals, and the other project used the GAN to come up with preliminary sketches for application icons, which were then finalized by the humans. Allowing non-programmers more control and intervention points is clearly desired. However, while it is relatively straightforward to expose a few more hyper-parameters into software such as RunwayML or 'friendly' code wrappers, the challenge is in providing non-experts with intuitions about the expected way to deal with these hyper-parameters.

Finally, a very active area of current computational research is how to make generative models such as GANs more 'controllable', using latent space algebra, latent space disentanglement and more (e.g., [23, 24, 25, 26]). We suggest that it is both interesting and important to see what happens when such tools are "unleashed" to the hands of non programmers. As we show here, they may discover new ways to improve results, which were not planned for by the algorithm designers. Notably, and contrary to the majority of algorithmic effort, students tried to obtain their goal by modifying the training data set – either by selecting a counter-intuitive pre-trained option, or by modifying their own datasets that were used for fine tuning the models.

Acknowledgments

The authors wish to thank all students taking part in the course, many of whom are mentioned in the text, and especially Gizem Odemir for also making some suggestions on the survey.

References

[1] A. Adadi, M. Berrada, Peeking inside the black-box: A survey on explainable artificial intelligence (xai), IEEE Access 6 (2018) 52138–52160.
[2] D. P. Kingma, P. Dhariwal, Glow: Generative flow with invertible 1x1 convolutions, in: Advances in neural information processing systems, 2018, pp. 10215–10224.
[3] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, A. Lerchner, Understanding disentangling in β-vae, arXiv preprint arXiv:1804.03599 (2018).
[4] M. A. Boden, Creativity, in: Artificial intelligence, Elsevier, 1996, pp. 267–291.
[5] M. H. Jarrahi, Artificial intelligence and the future of work: Human-ai symbiosis in organizational decision making, Business Horizons 61 (2018) 577–586.
[6] A. Bidgoli, E. Kang, D. C. Llach, Machinic surrogates: Human-machine relationships in computational creativity, arXiv preprint arXiv:1908.01133 (2019).
[7] J. K. Eshraghian, Human ownership of artificial creativity, Nature Machine Intelligence (2020) 1–4.
[8] C.-Z. A. Huang, H. V. Koops, E. Newton-Rex, M. Dinculescu, C. J. Cai, Ai song contest: Human-ai co-creation in songwriting, arXiv preprint arXiv:2010.05388 (2020).
[9] R. Louie, A. Coenen, C. Z. Huang, M. Terry, C. J. Cai, Novice-ai music co-creation via ai-steering tools for deep generative models, in: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, 2020, pp. 1–13.
[10] C. Oh, J. Song, J. Choi, S. Kim, S. Lee, B. Suh, I lead, you help but only with enough details: Understanding user experience of co-creation with artificial intelligence, in: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, 2018, pp. 1–13.
[11] P. Karimi, K. Grace, M. L. Maher, N. Davis, Evaluating creativity in computational co-creative systems, arXiv preprint arXiv:1807.09886 (2018).
[12] G. N. Yannakakis, A. Liapis, C. Alexopoulos, Mixed-initiative co-creativity (2014).
[13] P. Karimi, J. Rezwana, S. Siddiqui, M. L. Maher, N. Dehbozorgi, Creative sketching partner: an analysis of human-ai co-creativity, in: Proceedings of the 25th International Conference on Intelligent User Interfaces, 2020, pp. 221–230.
[14] M. T. Llano, M. d'Inverno, M. Yee-King, J. McCormack, A. Ilsar, A. Pease, S. Colton, Explainable computational creativity, in: Proc. ICCC, 2020.
[15] J. Zhu, A. Liapis, S. Risi, R. Bidarra, G. M. Youngblood, Explainable ai for designers: A human-centered perspective on mixed-initiative co-creation, in: 2018 IEEE Conference on Computational Intelligence and Games (CIG), IEEE, 2018, pp. 1–8.
[16] D. G. Novick, S. Sutton, What is mixed-initiative interaction, in: Proceedings of the AAAI spring symposium on computational models for mixed initiative interaction, volume 2, 1997, p. 12.
[17] K. Sims, Evolving virtual creatures, in: Proceedings of the 21st annual conference on Computer graphics and interactive techniques, 1994, pp. 15–22.
[18] A. Mordvintsev, C. Olah, M. Tyka, Inceptionism: Going deeper into neural networks (2015).
[19] A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, arXiv preprint arXiv:1511.06434 (2015).
[20] T. Karras, S. Laine, T. Aila, A style-based generator architecture for generative adversarial networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2019, pp. 4401–4410.
[21] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, T. Aila, Analyzing and improving the image quality of stylegan, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 8110–8119.
[22] O. Sendik, D. Lischinski, D. Cohen-Or, Unsupervised k-modal styled content generation, arXiv preprint arXiv:2001.03640 (2020).
[23] A. Jahanian, L. Chai, P. Isola, On the"steerability" of generative adversarial networks, arXiv preprint arXiv:1907.07171 (2019).
[24] E. Härkönen, A. Hertzmann, J. Lehtinen, S. Paris, Ganspace: Discovering interpretable gan controls, arXiv preprint arXiv:2004.02546 (2020).
[25] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, A. A. Efros, Generative visual manipulation on the natural image manifold, in: European conference on computer vision, Springer, 2016, pp. 597–613.

[26] T. Park, M.-Y. Liu, T.-C. Wang, J.-Y. Zhu, Gaugan: semantic image synthesis with spatially adaptive normalization, in: ACM SIGGRAPH 2019 Real-Time Live!, 2019, pp. 1–1.