VRIJE UNIVERSITEIT BRUSSEL

COMPUTER GENERATED CAR DESIGN

GANs and Computational Creativity

Lennert Bontinck

May, 2020-2021

Student number: 568702

Computer Science: AI

Abstract

This paper discusses the development and evaluation of a creative system capable of generating photorealistic novel car designs and modifying them. This system makes use of a pre-trained StyleGAN2 model (Karras et al., 2020) and a modified version of the GANSpace tool (Härkönen et al., 2020). The various components are discussed loosely based on the computational creativity (CC) system description paper by Ventura (2017). These components are also placed inside the creative systems framework (CSF) proposed by Wiggins (2006) to further clarify the creative aspects of the system.

This paper also aims to discuss the possibilities and shortcomings of generative adversarial networks (GANs) in the CC field. A more philosophical discussion is held to show such systems can indeed be creative rather than just generative. It is shown how conceptual space exploration tools such as the modified GANSpace tool can be used to combat the black-box problems with GANs. The need for CC specific internal evaluation and possible solutions are also briefly touched upon. The external evaluation performed aims to further defend the creativity of the made system and thus the viability of GANs as a creative system. The used tool for external evaluation was custom build for this project and is made available free to use and open source.

This paper was made as a requirement of the Computational Creativity course taught at the VUB. All source files for this project are available on GitHub (Bontinck, 2021). It is noted that this report is written using a modified version of the VUB based LaTeX template from De Smet (2020).

Contents

About the creative domain

In this chapter, the creative domain of car design is discussed. Relevant literature in the CC field and surrounding GANs are briefly summarised. The available resources for this project and the limits they bring with them are also touched upon.

1.1 Car design and Computational Creativity

The car industry is only just over a century old and has already evolved from a motorized luxury carriage for the rich to a multi-billion euro industry for the masses. In the last decade, the car industry has undergone major changes, with electrification and autonomous driving being the most prominent. Computer algorithms play a key role in these changes to ensure safe autonomous driving and optimal battery usage.

However, computer algorithms do much more for the car industry. They are used for crash test simulations (Fang et al., 2016), automotive aerodynamics (Dube & Hiravennavar, 2020) and more. This raises the question: if the industry uses so many computer-generated simulations and calculations for validating the design of cars, can't a computer generate a car design? This is a task that is gaining interest by big brands, especially in Formula 1 and hypercar design.

This paper explores that idea by building a creative system capable of generating photorealistic novel car designs and having control over the generated designs. It is noted that the output of this system is only that, a picture of a car. The true viability of the proposed car concerning legislation, safety and more are not taken into consideration. As is the case for architectural design, clothing design and many other design-related domains, car design is deemed a creative domain. Some might argue against this idea, especially if they're not interested in cars. Arguments could include that the limits put in place by legislation and the desire for reoccurring style traits of a brand limit the creativity in the domain. Whilst many attempts at formalising what is (not) creative, such as the important work by Boden (2004), have been made, it is still hard to objectively deem something creative. However, with numerous car museums, legendary car design brands as Pininfarina and culturally driven evolution in car design, the domain is deemed creative for this paper.

1.2 Relevant literature

Much interesting relevant literature exists. Some important papers on GANs and two relevant papers from the CC field are discussed in what follows. These papers give better insight into how the technology used for this paper's system works and its viability as a useful creative system in the domain.

1.2.1 Literature on GANs

Generative adversarial networks (GANs), first introduced by Goodfellow et al. (2014), are systems capable of generating output images by training both a generator and discriminator to play a form of cat-and-mouse game. More details on this idea are given later in this paper. Such networks are the state-of-the-art used for image generation and have been used for similar, non-scientific, projects by Soomar et al. (2020), Amitesh (2020), ODSC (2019) and more. Many different variants of GANs exist, with an impressive recent example being BigGAN-deep by Brock et al. (2019). Perhaps the most known GAN is StyleGAN by Karras et al. (2018), researchers at NVidia. It was used for a heavily media covered website that displays images of people who don't actually exist[1].

For the system of this paper, a pre-trained StyleGAN2 model is used. StyleGAN2 is a successor to the already mentioned StyleGAN (Karras et al., 2020). StyleGAN2 introduces many improvements over the basic GAN idea by making use of concepts from the style transfer literature. When trained on faces, the models seem to be capable of separating high-level attributes (e.g. orientation of face) and more low-level variations (e.g. presence of freckles). Other differences include the use of four distinct random noise vector inputs to intermediate layers as opposed to only one starting noise vector with basic GANs.

Since StyleGAN and StyleGAN2 are made with the idea of being able to learn the concepts of an image, such as the discussed orientation and freckles with face generation, tools to easily control these parameters are being developed. GANSpace by Härkönen et al. (2020) is one of them. GANSpace takes a trained model of either StyleGAN, StyleGAN2 or BigGAN and extracts interpretable controls for image synthesis of them. This is done by identifying important latent directions based on PCA analysis in the activation space of these GANs. The main contribution of this tool for this paper is that it allows exploration of a GANs conceptual space in an easy manner.

[1] https://thispersondoesnotexist.com/

1.2.2 Literature on car design in the CC field

Some non-scientific projects that use GANs to generate novel car designs were already mentioned. However, scientific papers on car design in the CC field aren't very common. A particularly interesting research paper worth mentioning is the one by Radhakrishnan et al. (2018). In their paper, a creative system is developed which takes a pen-and-paper sketch of a car and returns a graphical representation of what that car could look like in a more realistic representation. The authors of that paper deem their system creative since it is capable of generating images from angles of the car that are not visible in the input sketch.

Their paper also demonstrates that creative systems can not only be used to complete a creative task but also as a source of inspiration for human-made creative work. This is because, just like it is the case for this paper's system, their outputted images aren't meant to be viable or regulatory. They are meant to give an idea of what a car based on their input sketch could roughly look like. Moreover, their creative system proposes multiple variants based on the input sketch. This allows for the car designer to get inspirations for and an idea of what a finalised design could look like. The creative system is not made to replace a car designer's job, but rather to be a tool in optimising a car designer's workflow.

DARCI by Norton et al. (2013) is a well-known system in the CC field that produce images through creative means. DARCI is accepted as being a creative system in the field that makes use of neural networks (NNs). Its use of NNs is important when discussing the viability of using GANs in the CC field later in this paper.

1.3 Available resources

GANs require a lot of images, often in the millions, to generate pleasing results. Some of the pre-trained StyleGAN2 models made available by NVidia used over 20 million images for certain configurations (Karras et al., 2020). Whilst it would be possible to create a web-scraper that gathers such a quantity of images from car auction websites, it isn't viable for this project. This would mean an existing database of car images would need to be used for training the StyleGAN2 model. One such database might be the LSUN-Stanford Car one by Kramberger and Potočnik (2020). However, the amount of time required to train a StyleGAN2 model with this amount of data would be weeks, if not months with the computational power available for this paper. Using fewer images and/or epochs would most likely result in non-viable results. This is one of the reasons a pre-trained model is used.

For this paper, access to numerous experts in the car industry with excellent knowledge of existing car brands and models was available. These people are seen as juries with expertise. This helped tremendously to determine the P-creativity and H-creativity of the system, which is further discussed in the evaluation chapter of this paper.

GANs and creativity

An important, but often difficult to answer, question in the CC field is whether a system is creative or simply generative. To defend the claim of GANs being capable of becoming creative systems, the main idea behind them is first explained in a simplified manner. The black box problem of GANs is discussed together with the challenges it forms. This chapter concludes with a more philosophical take on why GANs can be creative systems.

2.1 Main idea behind a GAN

Figure 2.1 shows a visualisation of the different components of a GAN. It becomes visible that there are two systems at play, a discriminator and a generator. The discriminator has access to the database containing images of the concept the GAN should generate new instances for. Its job is to either accept or reject an incoming image. An image that is accepted can be seen as one that the discriminator finds to be of the concept it has learned and thinks it is not computer-generated. In some instances, the discriminator is a continuous learning algorithm. The generator learns by starting from a random noise vector and transforming it until it receives something accepted by the discriminator. The main differences between different GAN technologies are the way how they implement the generator.

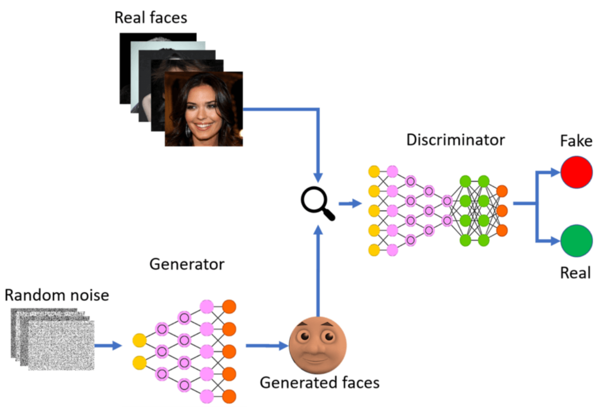


Figure 2.1: Basic idea of a GAN.
Figure by Kulpati (2019)

2.2 Black box problem

One issue with GANs is that they almost always make use of deep NNs for both their generator and discriminator. A deep NN is known to be a black-box model. This means that there is no meaningful insight on how the model works, it is just a bunch of mathematical manipulations a human cannot reason about. This makes it hard to state that a deep NN has learned something. One can only evaluate if the output is what is desired. This introduces the risk of claiming that you think a deep NN has learned something whilst in reality it might have learned something completely different which happens to give the same desired output by chance. This is an important issue for AI in general. An academic example that is often given to demonstrate this point is one where a tank recognition NN didn't learn to differentiate trees and tanks but rather the time of the day, which happened to correspond with the presence of tanks for the training and test samples. Whilst this example demonstrates the issue at hand, it is noted that it could be an urban legend as discussed in a blog post by Gwern Branwen [1].

2.3 GANs as creative systems

In the CC field, understanding the process for achieving an output that might be deemed creative by humans is an important manner in deeming a system creative. This makes algorithms with the black box problem such as GANs a hard case for creative systems. However, the discussed DARCI system by Norton et al. (2013) shows that systems using NNs can indeed be deemed creative if defended appropriately. One of the convincing strengths of DARCI is that it gives a natural language description of the steps it takes to generate an image. These are things such as explaining what image of the training set it relates to and what the labels for that known image are. These descriptions aim to give an insight into the process of the creative system to try and tackle the black box problem. However, DARCI does not know the semantic meaning of those descriptions and labels. All of the natural language descriptions and labels were provided by a human as training data and statistically processed by the system. It is even very likely that the internal representation of those strings is numerical through encoding. Knowing this, the natural descriptions it delivers are likely to give the false perception that the system has a sense of semantic reasoning, which it does not. It also gives a wrong idea of how the generative process of the system works.

Not all is bad though, as is mentioned in the paper on how to build a CC system by Ventura (2017), some argue that not knowing what aesthetic the system is using is a positive advance as it gives more freedom and thus enhances creativity. Whilst it would be easy to just agree with this as a defence on why explainability shouldn't be of huge concern for the CC field, there are some issues with this. First of all, Ventura is a co-author of the older DARCI system. Since DARCI uses NNs, Ventura couldn't suddenly state explainability are a must for creative systems, as that would render DARCI a non-creative system. His use of GANs as examples in the paper further shows his stance on this topic. However, one of the three characteristic Ventura gives to computationally creative agents is intentionality. To defend intentionality correctly, a certain idea of how the generative process works needs to be present. Explainability is also something that is desired for scientific fields in general.

[1] https://www.gwern.net/Tanks

So, where does this leave GANs as (part of) a creative system? As it turns out, the days of true black-box models, where no analysis of the learned concept was performed seem to have passed. This is mainly due to new regulations being hard on the explainability of AI algorithms. Since deep NNs are often the desired solution for AI problems, much research in making them explainable is being done.

Papers such as the one by Bau et al. (2020) have an interesting approach to enhance explainability. They analyse the meaning of individual units of a deep NN by disabling or boosting their output. From this, they conclude multiple things. One example is the detection of units that are responsible for generating trees, as disabling them drastically lowers the generation of trees and vice versa. These approaches are steps in making it more defensible that a deep NN has indeed learned certain concepts. This leads to why tools such as GANSpace can aid in the explainability of a GAN. It also gives an insight into the generation process of the GAN and thus the use of GANs in the CC field becomes more viable. If it is possible to determine components within one or more layers of the GAN to be responsible for the creation of distinct concepts in hundreds of samples, can it then not be claimed the system has learned that concept? When does luck become skill? On the GitHub repository for this paper, multiple examples of changing distinct concepts through GANSpace can be found. Some are also discussed in the evaluation chapter of this paper and shown in figure 5.3. From this, it becomes clear the system does have an explainable generation process that consists of generating multiple distinct concepts to form a final image.

Once the above statements about the GAN's generation procedure are accepted, it's an easier step towards claiming them creative systems. In the discussed internal working of a GAN, it is visible that internal evaluation is a crucial part of the design already. GANs are also more than just an optimisation problem, as there is no "best image to generate", much as there is no "best artwork to generate". Since the generator doesn't have direct access to the dataset and uses random noise vectors as its input it becomes clear it is creating things from the knowledge it has learned through the discriminator. From the above statements, it becomes clear that knowledge is indeed meaningful concepts rather than random values. The training loop then limits the conceptual space of the generator from all possible combinations of pixels to all images accepted by the discriminator. This process corresponds to the generator becoming better at understanding the different concepts a car design requires.

One of the only challenges remaining to deem GANs creative is whether or not the created images are different enough from existing cars. A reverse image search analysis, such as the ones used by Google reverse image search, can be used to ensure novelty. Most of such state-of-the-art algorithms for this task also use deep NNs. A recent paper by Diyasa et al. (2020) discusses how such behaviour could be reached by using a pre-trained convolutional NN. It is noted that adding such an evaluation metric would only ensure P-creativity, as it only has access to the training images and not all cars ever made. It is also important to remember from the domain explanation that similarity to existing models is expected and often even desired in car design. Thus, for this specific domain, there should possibly even be a lower bound for this similarity metric. Having hyper-parameters for these thresholds is recommended. With this final required component of the system in place, it can be seen that GANs can indeed be creative systems.

Design of the system

Backed by relevant literature and a more philosophical take on how GANs can be deemed creative systems, the design of the system is discussed in this chapter. The working of the various components is presented loosely based on the CC system description paper by Ventura (2017). These components are also placed inside the creative systems framework (CSF) proposed by Wiggins (2006) to further clarify the creative aspects of the system.

3.1 Overview of the system

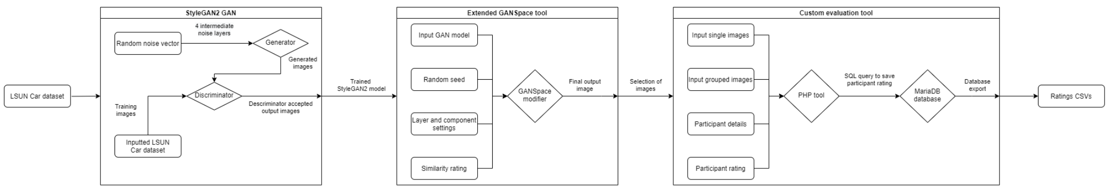


Figure 3.1: High level overview of whole creative system's pipeline including external evaluation.

The whole pipeline of the developed system is visualised in figure 3.1. For this paper, the first step of training the StyleGAN2 model is bypassed by using a pre-trained one. The StyleGAN2 car pre-trained model with configuration f is used. This model generates images of cars at a resolution of 512 by 384 pixels. It used the LSUN Car dataset for training, which is part of the discussed LSUN-Stanford Car dataset by Kramberger and Potočnik (2020). It was the highest resolution pre-trained car model at the time of writing. With the use of an NVIDIA DGX-1 with 8 Tesla V100 GPUs, it took 4 days and 18 hours to train (Karras et al., 2020). The different components of the extended GANSpace tool, as well as the custom evaluation tool, are discussed later in this paper.

It becomes clear all of the steps and components proposed by Ventura (2017) are in place. As already discussed, the domain of car design is chosen. Car design is represented by a phenotypic representation of images of cars. Internally the genotypic representation of these images is a numerical vector representing the images. Their exact representation is StyleGAN2 specific.

The used data is the LSUN Car dataset which through conversion to the genotypic representation is used as the knowledge base by the discriminator of the GAN. The generator function is the generator of the GAN which makes use of four intermediate random noise vectors as ”input” as well as the learned style layers and more. More technical details on this can be found in the StyleGAN2 paper (Karras et al., 2020). There are multiple aesthetic measures. These include the inclusion of required car components, recognisability of existing car brand styling traits and more. These are mainly taken care of by the discriminator which serves as the value measure. The similarity rating, which is part of the extended GANSpace tool, serves as the novelty measure. The external evaluation serves as the phenotypic evaluation.

3.2 Defending the design decisions

Due to the discussed available resources being rather limited, a pre-trained StyleGAN2 model was the only feasible choice. As is discussed later in the paper where the internal evaluation is explained, the similarity rating is done manually due to limited resources as well. However, it can be made automatic, which is also discussed later in this paper. The made design decisions don't limit the creative system of this paper but give rise to interesting future extensions, which are discussed at the end of this paper.

3.3 Putting it in terms of the CSF

A presentation where the system is situated in terms of the CSF was given at the VUB and the used slides are available under the presentations folder of the GitHub repository for this paper (Bontinck, 2021). A summary of the contents is given below.

- The universe U is Technically all RGB combination of pixels. For the generator, this boils down to all images deemed “real” by the discriminator. When assuming the discriminator does its job, the universe thus consists of all images of realistic-looking cars.
- The conceptual space C is the set of all images the generator can make based on all possible noise vectors using its latest transformers. It is clear that $C \subset U$
- Remember from the CSF that $C = [[R]](U)$. This means the rules R constraining the space are the same rules defining the state of the generator.
- The rules T are those that introduce randomness and noise as restrictions on the latent spaces. At a high level, the extended GANSpace tool uses these rules to explore the conceptual space.
- The rules E are those that define the discriminator as it can be used to assess the quality of the generated image. The similarity rating also belongs to these rules. To keep the language L unchanged, it should be implemented in the same language being a combination of Python and C++.
- Since $F^{\diamond}$ is limited by the output of images having 512 by 384 pixels it is finite. This means $e_c =< R, T, E >^{\diamond} (\{T\})$ is also finite. Since GANSpace could bypass some of StyleGAN2's rules it is possible that $e_c \not\subset C$.

Implementing the creative system

This chapter goes over the technical details of the implementation of the system. Firstly, the used pre-trained StyleGAN2 model (Karras et al., 2020) is discussed. The issues with and extensions made to the GANSpace tool (Härkönen et al., 2020) are also explained here. Since the GANSpace tool lacks documentation and support, special attention has been taken to ensure reproducibility.

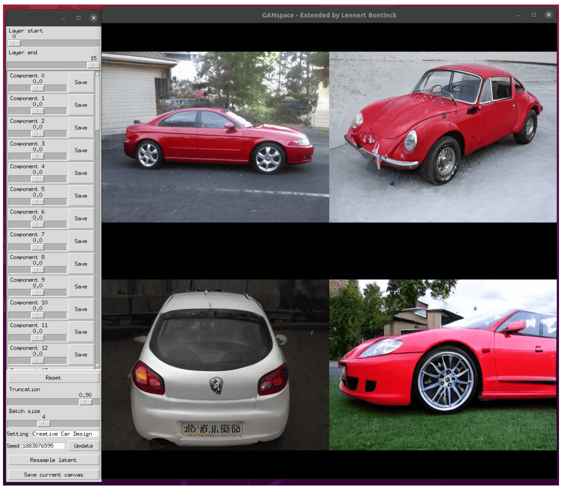
4.1 Using StyleGAN2 model

Using a pre-trained StyleGAN2 model is easy and doesn't require an extraordinary amount of computational power. As StyleGAN2 is created by researchers at NVidia, it's recommended to have an NVidia GPU, especially one with CUDA support. These are widely available and the system used for this project had a relatively old GTX 970 with 4 GB of VRAM. With just a few lines of Python code and the download of a pre-trained model, which is about 1 GB in size, the GAN is loaded in and starts generating images. The generation of a singular image takes seconds at most for the GTX 970 system. Google's Colaboratory Notebooks are also capable of doing just this with the free tier. Since downloading and starting the pre-trained model is taken care of automatically by the extended GANSpace tool, no further instructions are given.

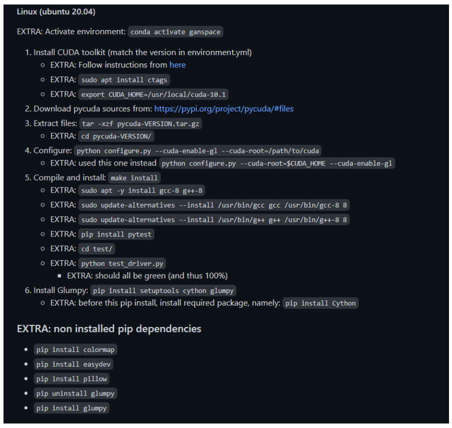
4.2 Extending the GANSpace tool

Whilst GANSpace is an incredibly easy tool to gain interpretable control over a GAN with, it does have some shortcomings and flaws. As a starter, the documentation isn't up to par and due to the low popularity, troubleshooting isn't an easy task. The installation guide at the time of writing lacks crucial steps such as specifying all required dependencies and more. Support for anything other than Linux distributions is poor. Because of this, it was chosen to create a fresh install of Ubuntu 20.04 and document all steps required to get the tool to work. These steps and much more documentation are available on the GitHub repository of this project (Bontinck, 2021). The new setup guide is available under the GANSpace folder as SETUP.md, it is also published to the issues page of the GANSpace tool[1]. The used Pycuda folder is also made available.

[1]https://github.com/harskish/ganspace/issues/49



(a) Extended GANSpace tool



Linux (ubuntu 20.04)

EXTRA: Activate environment: `conda activate ganspace`

1. Install CUDA toolkit (match the version in environment.yml)
   - EXTRA: Follow instructions from here
   - EXTRA: `sudo apt install ctags`
   - EXTRA: `export CUDA_HOME=/usr/local/cuda-10.1`
2. Download pycuda sources from: https://pypi.org/project/pycuda/#files
3. Extract files: `tar -xzf pycuda-VERSION.tar.gz`
   - EXTRA: `cd pycuda-VERSION/`
4. Configure: `python configure.py --cuda-enable-gl --cuda-root=/path/to/cuda`
   - EXTRA: used this one instead `python configure.py --cuda-root=$CUDA_HOME --cuda-enable-gl`
5. Compile and install: `make install`
   - EXTRA: `sudo apt -y install gcc-8 g++-8`
   - EXTRA: `sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-8 8`
   - EXTRA: `sudo update-alternatives --install /usr/bin/g++ g++ /usr/bin/g++-8 8`
   - EXTRA: `pip install pytest`
   - EXTRA: `cd test/`
   - EXTRA: `python test_driver.py`
     - EXTRA: should all be green (and thus 100%)
6. Install Glumpy: `pip install setuptools cython glumpy`
   - EXTRA: before this pip install, install required package, namely: `pip install Cython`

EXTRA: non installed pip dependencies

- `pip install colormap`
- `pip install easydev`
- `pip install pillow`
- `pip uninstall glumpy`
- `pip install glumpy`

(b) Exhaustive documentation

Figure 4.1: Screenshot of the extended GANSpace tool and documentation.

Besides this, a simple script called rungan.sh is made, which upon execution initializes the GANSpace environment exactly as it was used for this paper. After successfully running this script, a screen as shown in figure 4.1a should be presented. Note the visual queues such as the setting field being set to 'Creative Car Design' and the title of the window being custom for the project. The most important added feature is the possibility of saving the canvas. This works for all possible batch sizes. The code responsible for generating different images after resampling the latent space is also discussed inside the exploring_GANSpace.md file under the GANSpace folder. This is important for the similarity measure as is discussed in the evaluation chapter of this paper.

4.3 Reproducibility

In contrast with the original GANSpace tool documentation, this project contains all required details on downloading and running all required dependencies and code. The custom made script that was discussed allows for a one-line command to initialize the GANSpace tool identically to the one used. It downloads the discussed pre-trained StyleGAN2 model automatically. All of the settings for generated images and found modifications are also written down. These are documented in the exploring_GANSpace.md file under the GANSpace folder as well as the README.md of the generated images folder. This means that by simply setting the seed and changing the corresponding sliders as documented, identical results can be achieved.

Evaluation

Perhaps the most important part of a creative system is evaluation, both internally as external. This chapter goes over both. Due to the limited resources, internal evaluation is non-optimal but solutions are proposed. Some of the reoccurring issues with the system's output are also discussed. A heavy focus was put on the external evaluation and a custom build evaluation tool was made. How this tool works and the results of the external evaluation are discussed here.

5.1 Internal evaluation

As was already discussed, the discriminator is responsible for automatic internal evaluation. The generator can use it to assess the quality of the generated image by either being accepted as real or not. However, the point was raised that it is not possible to validate whether the generated images differ enough from existing cars. If the discriminator is too strict, the generator can end up generating nearly identical images to the one in the training set. Such a system would produce incredibly realistic results but is hard to call creative. Currently, StyleGAN2 and other state-of-the-art models don't seem to have a metric in place to strictly evaluate this behaviour.

However, as was already discussed, a solution for this problem exists in the form of a reverse image search analysis. Due to limited resources for this paper, this was not implemented. However, the documentation on where the generation of new images occurs in the extended GANSpace tool is discussed. This allows for future extensions where an automatic evaluator can be made to skip the output of the StyleGAN2 model that resembles a car from the training set too much. It is noted that this ideology was used in the selection of images for evaluation, where three domain experts sat together to discuss which cars they recognized. These experts were a Peugeot car mechanic, a Mercedes sales manager and a Honda employee. From this, a selection similar to the use of such an evaluation metric was made. It is also important to note that providing this metric inside GANSpace would make it a wrapper function, meaning, the actual training of the model remains unchanged. In an ideal world, this evaluation metric would be built into the used GAN technology.

Finally, the extended GANSpace tool helped validate the GAN has indeed learned concepts related to car design by using the same reasoning as the discussed paper by Bau et al. (2020). The generation process thus consists of making an image that contains learned concepts of a car. Since the generated car designs are different enough from existing cars, they can be deemed creative in a similar manner Radhakrishnan et al. (2018) deemed their system creative.

(a) Double front car

(b) Missing piece in bed

(c) Attempt at text

Figure 5.1: Multiple images generated by the system that demonstrate some of its flaws. Settings and more info available on GitHub (Bontinck, 2021).

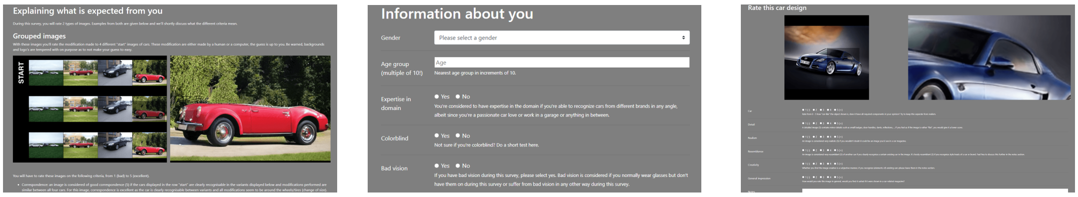
5.2 Challenges for the GAN

Whilst analysing the output of the GAN, it became visible the GAN had some reoccurring issues with its generated images. One of these issues is what the documentation available on GitHub calls the 'challenging' angle. When looking at the car from an angle in between front and side, the GAN seems to confuse what is front and back. This results in realistic images of cars that have two fronts or two backs. One such example is shown in figure 5.1a.

Another challenge seemed to be that the GAN didn't always seem to generate the car in all dimensions. This is shown in figure 5.1b, where a piece of the bed on the pick-up and convertible crossover is missing and replaced by grass. Including images where such dimensions are missing might be interesting for the external evaluation to get an idea if participants notice it.

The final reoccurring issue that is worth mentioning, is the system's poor performance in generating realistic backgrounds. Whilst generating static backgrounds such as grass and trees was pretty successful most of the time, generating more complex environments often failed. It could be argued this is a good thing, as it means the discriminator and generator have a focus on the realism of the car, which is the purpose of this system. One of the most interesting phenomenons with background generation was the system's attempt at generating text. An interesting example of this is shown in figure 5.1c.

Other reoccurring anomalies from regular car designs were also noticed. One such example is the fact that the headlight design was different on the other side of the car. Whilst this is (almost) never found in car designs presents, it's not seen as an "issue of the GAN". Besides this, the generated cars are often relatively symmetrical. It is noted that the GAN also produced some outlandish results from time to time. However, this was less than 10% over more than 500 samples. One such artefact is shown in figure 5.4a.



(a) Explanation of survey (b) Personal information (c) Single image rating

Figure 5.2: Screenshots of the created external evaluation tool.

5.3 Making a suitable external evaluation tool

This paper aimed to make a creative system capable of generating photorealistic novel car designs. The external evaluation helps in proving this goal is reached. To ensure proper evaluation, a custom made PHP based survey tool is made. This tool is based on a previous project from the author of this paper (Bontinck, 2019). Some screenshots of the tool, which was made available online[1], are shown in figure 5.2. The flow of the system is described below.

- Show participant what personal information will be collected and what the survey is about.
- Show a more detailed explanation of all fields that need answering (see figure 5.2a):
  - Explains some images are made by a computer algorithm and others by a human through photo editing software.
  - Explains the backgrounds and logos have been tampered with on purpose to make the above choice harder. This is done due to the issues the GAN has with more complex backgrounds as described earlier.
- Ask the user some personal information (see figure 5.2b):
  - Gender (optional) and age in multiple of 10. The latter is done to improve anonymity.
  - Expertise: Participant is considered to have expertise in the domain if he is capable of recognizing cars from different brands from any angle.
  - Whether the participant is colour blind or has other vision issues when taking the survey (e.g. not wearing glasses).

[1]https://www.lennertbontinck.com/creative-car-design-survey/

- Show two grouped images and four single images. These are selected at random from the evaluation set.
  - Grouped images are images where the GANSpace tool is used to perform modifications. The following criteria are asked using a Likert scale from one to five:
    * Correspondence: an image is considered of good correspondence (5) if the cars displayed in the row "start" are recognisable in the variants displayed below and modifications performed are similar between all four cars.
    * Realism: an image is considered very realistic (5) if it could be an image from a car magazine.
    * Creativity: Subjective measure, recognition of (elements of) existing cars can influence this.
    * Made by: Whether participant thinks the modifications are made by a human or a computer.
    * Notes: field for the participant to leave notes such as recognized cars.
  - Single images are images straight out of the StyleGAN2 model, taking into account the selection done using the similarity measure. The following criteria are asked using a Likert scale from one to five:
    * Car: how "car-like" the object shown is, does it have all required components etc.
    * Detail: a detailed image (5) contains minor details such as small badges, door handles, dents, reflections, ... If the image is rather "flat", it would receive a lower score.
    * Realism: same ideology as before.
    * Resemblance: an image is considered very resemblant (5) of another car if the participant recognizes a certain existing car in the image. It's loosely resemblant (3) if he recognizes the style treats of a car or brand.
    * Creativity: same ideology as before.
    * General impression: how the participant would rate this image in general.
    * Made by: same ideology as before.
    * Notes: same ideology as before.
- Show the participant a screen admitting he's been lied to and all images have been computer-generated. Ask whether or not the participant thinks he was biased.
- Similarly show the other evaluation images.

The tool is made available under the GPL V3 license and is available with documentation on the GitHub repository of this project (Bontinck, 2021). On the same GitHub page, all used images for the external evaluation and the documentation to reproduce them is also available. Some of the images generated through GANSpace modifications were minor changes such as different wheel designs whilst others were major changes such as making a car look more sporty. These are displayed in figure 5.3. The single images contained various examples, from super realistic examples to creative artefacts of the system. Some are shown in figure 5.4.

(a) Minor edits

(b) Major edits

Figure 5.3: Some of the grouped images used for external evaluation. Note that the textual description was removed when shown to the participant.

(a) Creative artefact

(b) Very realistic

(c) Challenging angle

(d) Double front

(e) Missing piece

(f) Mercedes/BMW merge

Figure 5.4: Some of the single images used for external evaluation.

5.4 External evaluation results

Conclusions

With all major parts of the creative system discussed, a reflection on the system is given. A short rehearsal on what GANs would need to be widely adopted and accepted as creative systems is given. The added value of this paper and interesting future work are also mentioned.

6.1 Reflection on the designed system

6.2 Added value

6.3 Interesting future work

References

Amitesh, O. (2020). *Gan to generate images of cars*. Retrieved March 8, 2021, from https://medium.com/swlh/gan-to-generate-images-of-cars-5f706ca88da

Bau, D., Zhu, J.-Y., Strobelt, H., Lapedriza, A., Zhou, B., & Torralba, A. (2020). Understanding the role of individual units in a deep neural network. *Proceedings of the National Academy of Sciences*, *117*(48), 30071–30078. https://doi.org/10.1073/pnas.1907375117

Boden, M. (2004). *The creative mind: Myths and mechanisms*. Routledge. https://books.google.be/books?id=6Zkm4dz32Y4C

Bontinck, L. (2019). *Je kijkt er naar, maar ziet het niet: Datacompressie principes — ontstaan en uitdagingen — implementaties — afbeeldingscompressie: Png — jpeg — jpeg2000 — webp — heif — videocompressie: H.264-avc — h.264-svc — h.265/hevc — av1* [Available under "evaluatietool"]. Retrieved March 21, 2021, from https://github.com/pikawika/bachelorproef-compressie

Bontinck, L. (2021). *Computational creativity project* [GitHub commit: 450e822...]. Retrieved March 7, 2021, from https://github.com/pikawika/VUB-CC-Project

Brock, A., Donahue, J., & Simonyan, K. (2019). Large scale gan training for high fidelity natural image synthesis.

De Smet, R. (2020). *Vub latex huisstijl* [GitHub commit: d91f5579...]. Retrieved November 2, 2020, from https://gitlab.com/rubdos/texlive-vub

Diyasa, I. G. S. M., Alhajir, A. D., Hakim, A. M., & Rohman, M. F. (2020). Reverse image search analysis based on pre-trained convolutional neural network model. *2020 6th Information Technology International Seminar (ITIS)*, 1–6. https://doi.org/10.1109/ITIS50118.2020.9321037

Dube, P., & Hiravennavar, S. (2020). Machine learning approach to predict aerodynamic performance of underhood and underbody drag enablers. *SAE Technical Paper Series*. https://doi.org/10.4271/2020-01-0684

Fang, J., Sun, G., Qiu, N., Kim, N. H., & Li, Q. (2016). On design optimization for structural crashworthiness and its state of the art. *Structural and Multidisciplinary Optimization*, *55*(3), 1091–1119. https://doi.org/10.1007/s00158-016-1579-y

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, 2672–2680.

Härkönen, E., Hertzmann, A., Lehtinen, J., & Paris, S. (2020). Ganspace: Discovering interpretable gan controls.

Karras, T., Laine, S., & Aila, T. (2018). A style-based generator architecture for generative adversarial networks. *CoRR*, *abs/1812.04948*. http://arxiv.org/abs/1812.04948

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., & Aila, T. (2020). Analyzing and improving the image quality of StyleGAN. *Proc. CVPR*.

Kramberger, T., & Potočnik, B. (2020). Lsun-stanford car dataset: Enhancing large-scale car image datasets using deep learning for usage in gan training. *Applied Sciences*, *10*(14). https://doi.org/10.3390/app10144913

Kulpati, S. (2019). *A brief introduction to gans*. Retrieved May 8, 2021, from https://medium.com/sigmoid/a-brief-introduction-to-gans-and-how-to-code-them-2620ee465c30

Norton, D., Heath, D., & Ventura, D. (2013). Finding creativity in an artificial artist. *The Journal of Creative Behavior*, *47*, 106–124. https://doi.org/10.1002/jocb.27

ODSC, T. (2019). *Using gans to generate images of race cars*. Retrieved March 8, 2021, from https://opendatascience.com/using-gans-to-generate-images-of-race-cars/

Radhakrishnan, S., Bharadwaj, V., Manjunath, V., & Srinath, R. (2018). Creative intelligence - automating car design studio with generative adversarial networks (gan). In A. Holzinger, P. Kieseberg, A. M. Tjoa, & E. Weippl (Eds.), *Machine learning and knowledge extraction* (pp. 160–175). Springer International Publishing.

Soomar, A., Balaji, R., McCray, R., Hung, E., Guggari, S., & Cleaver, N. (2020). *Using neural nets to design cars*. Retrieved March 8, 2021, from https://medium.com/@alisoomar/adversarial-networks-for-car-image-generation-9bdf5977bec8

Ventura, D. (2017). How to build a cc system. *ICCC*.

Wiggins, G. A. (2006). A preliminary framework for description, analysis and comparison of creative systems. *Know.-Based Syst.*, *19*(7), 449–458. https://doi.org/10.1016/j.knosys.2006.04.009