



Universidade Federal de Pernambuco
Centro de Informática

Graduação em Ciência da Computação

**Utilizando redes adversárias generativas
(GANs) como agente de apoio à inspiração
para artistas**

Claudio Carvalho de Oliveira

Trabalho de Graduação

Recife
Abril / 2021

Universidade Federal de Pernambuco
Centro de Informática

Claudio Carvalho de Oliveira

**Utilizando redes adversárias generativas (GANs) como
agente de apoio à inspiração para artistas**

Trabalho apresentado ao Programa de Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Filipe C. A. Calegario

Recife
Abril / 2021

Dedico este trabalho à minha gata, Kita.

Agradecimentos

Gostaria de agradecer primeiramente a todos os grandes amigos que conheci durante minha graduação, sem os quais esta jornada acadêmica jamais poderia ser tão memorável. Com eles, as noites viradas no CIn movidas a K Energy Drink™ para alcançar o prazo de entrega de projetos extensos tornaram-se momentos bem-humorados e divertidos, criando assim uma motivação para superar qualquer dificuldade.

Sou extremamente grato também ao grupo PET-Informática e seus integrantes por me proporcionar uma experiência sem igual de realizar projetos de pesquisa, ensino e extensão na universidade.

Não poderia deixar de mencionar também, claro, todos os excelentes professores que me proporcionaram com uma quantia inestimável de conhecimento. Em especial, agradeço ao Prof. Dr. Filipe Calegario por me orientar neste trabalho e aos Profs. Drs. Juliano Iyoda e Leopoldo Teixeira por me permitirem fazer parte das monitorias de suas disciplinas, nas quais aprendi imensamente.

Por fim, agradeço principalmente à Jennyfer, a palmeira imperial adquirida durante uma pacata madrugada na Avenida Caxangá, levada para o CIn, e posteriormente lá plantada. Sua presença sem dúvidas motivou-me imensamente durante a graduação e me deu forças para seguir em frente.

Muito obrigado a todos!

*This was a triumph.
I'm making a note here:
HUGE SUCCESS.*

— GLaDOS

Resumo

Nos últimos anos, as redes adversárias generativas (GANs) vêm apresentado promissores resultados no campo da síntese de imagens. Seu uso para a produção de artes visuais, porém, não é acessível para muitos artistas devido aos requisitos de conhecimento técnico e hardware necessários. Este trabalho propõe e desenvolve uma interface simplificada para o treinamento destas redes, assim como um novo agente de apoio à inspiração para uso artístico por meio da exploração do espaço latente da rede. Para isto, foram utilizadas a rede StyleGAN2-ADA e a plataforma Google Colaboratory para criar um ambiente de execução interativo em nuvem. Os resultados obtidos em experimentos com artistas apontam um bom potencial para as soluções apresentadas, apesar de detectar também alguns problemas de experiência de usuário decorrentes da plataforma escolhida. É proposta ainda uma interface ainda mais acessível como solução a ser estudada para trabalhos futuros.

Palavras-chave: Redes Adversárias Generativas, GAN, Artes Visuais, Acessibilidade, Espaço Latente, StyleGAN, Aprendizado de Máquina

Abstract

In recent years, generative adversarial networks (GANs) have shown promising results in the field of image synthesis. Their usage for visual arts, however, is not accessible for many artists due to the technical knowledge and hardware requirements. This work proposes and develops a simplified interface for the training of these networks, as well as a new inspiration support agent for artistic use by exploring the network's latent space. To achieve this, the StyleGAN2-ADA network and the Google Colaboratory platform were used to create an interactive runtime environment in the cloud. The results obtained in experiments with artists show great potential for the presented solutions, although it also detects some user experience problems arising from the chosen platform. An even more accessible interface is also proposed as a solution to be studied for future works.

Keywords: Generative Adversarial Networks, GAN, Visual Arts, Accessibility, Latent Space, StyleGAN, Machine Learning

Sumário

1	Introdução	1
1.1	Contexto	1
1.2	Objetivos	2
2	Redes Adversárias Generativas	3
2.1	Funcionamento da GAN	3
2.2	StyleGAN	6
3	Aplicações Artísticas das GANs	9
3.1	Exemplos de Obras	9
4	Preparação do Conjunto de Dados	16
4.1	Seleção de Imagens	16
4.2	Pré-Processamento	18
5	Treinamento do Modelo	21
5.1	Execução	21
6	Exploração do Espaço Latente	23
6.1	Metodologia	23
6.2	Aplicação Desktop	24
6.3	Notebook para Google Colaboratory	26
7	Avaliação	28
7.1	Metodologia	28
7.2	Resultados	29
7.3	Discussão	30
8	Conclusão	33
8.1	Trabalhos Futuros	33

Lista de Figuras

2.1	Exemplo de possíveis pontos em um espaço latente simplificado bidimensional e as imagens que estes poderiam gerar.	4
2.2	Interpolação de pontos em um modelo de faces humanas.	5
2.3	Diagrama taxonômico das GANs sintetizadores de imagem.	6
2.4	Imagens de pessoas que não existem geradas pelo site This Person Does Not Exist.	7
2.5	Comparação entre a arquitetura de uma GAN padrão e de uma StyleGan.	8
3.1	Coleção de 2018 da AICAN.	10
3.2	Obra do Ganvas Studio exposta em parede.	10
3.3	Obra do Neural Zoo de Sofia Crespo.	11
3.4	Quadros da produção <i>Learning to See</i> de Memo Akten.	11
3.5	Exemplares gerados pela Emotional GAN utilizando análise de sentimento como entrada.	12
3.6	Obras de Helena Sarin, que experimenta com diferentes implementações de GANs para cada uma de suas obras.	12
3.7	Unknown Departure, uma obra criada combinando duas GANs.	13
3.8	Dois momentos distintos de uma exploração audiovisual do espaço latente de uma StyleGan.	13
3.9	Suéter produzido pela HATRA (ハトラ) a partir de uma GAN treinada com imagens de pássaros.	14
3.10	Exemplares gerados pelo Toonify Yourself, uma rede StyleGan que utilizou <i>transfer learning</i> a partir de imagens de faces humanas para faces de personagens de desenhos animados do cinema.	14
3.11	Processo de treinamento de uma GAN realizado por Yuguang Zhang ao gravar a si mesmo dormindo para captar o impacto de suas posições no lençol.	15
4.1	Imagens aleatórias geradas por um modelo treinado com exemplares do livro <i>Kunstformen der Natur</i> , tendo usado como base um <i>transfer learning</i> a partir de um modelo de faces humanas.	17
4.2	Visão geral da plataforma Google Colaboratory.	19
4.3	Comparação entre imagens antes e depois da etapa de pré-processamento.	20
5.1	Exemplo de um conjunto de imagens geradas a partir de um modelo intermediário.	22
5.2	<i>Logs</i> de execução de um treinamento.	22

LISTA DE FIGURAS

x

6.1	Evolução de uma população utilizando o algoritmo genético em uma rede treinada com obras de Frea Buckler.	24
6.2	Interface da aplicação desktop com exemplares gerados a partir de uma rede treinada com desenhos de figuras humanas.	25
6.3	Interface da aplicação desktop com exemplares gerados a partir de uma rede treinada com obras de Frea Buckler.	25
6.4	Interface executando em um <i>notebook</i> no Google Colaboratory com exemplares gerados a partir de uma rede treinada com obras de arte moderna.	26
6.5	Interface executando em um <i>notebook</i> no Google Colaboratory com exemplares gerados a partir de uma rede treinada com faces de personagens de anime.	27
7.1	Resultados médios obtidos pelo User Experience Questionnaire.	30

CAPÍTULO 1

Introdução

Nos últimos anos, as redes adversárias geradoras (GANs) [26] vêm recebendo cada vez mais atenção pela sua capacidade de atingir resultados excepcionais na criação de modelos geradores para os mais diversos domínios. Em especial, a síntese de imagens artificiais está constantemente em destaque devido aos avanços possibilitados pelo uso de tais redes.

A arquitetura da GAN possibilita esse aprendizado ao colocar duas redes com propósitos distintos para concorrer uma contra a outra em um jogo de soma-zero. A primeira dessas gera conteúdo (inicialmente aleatório) com o intuito de que este pareça ser pertencente ao conjunto de dados original, enquanto a segunda avalia o que foi gerado e tenta distinguir dados reais e falsos. Após cada ciclo da competição, ambas aprendem por meio de um mecanismo de retropropagação.

No campo da geração de imagens, temos hoje uma série de diferentes implementações de GANs, cada uma utilizando diferentes técnicas e heurísticas especializadas para resolver sub-problemas diferentes [43]. Dentre estas, a StyleGAN da Nvidia vem recebendo grande destaque como um exemplar do que já podemos alcançar com o atual estado da arte em quesito de imagens em super resolução. Esta rede utiliza-se do poder de processamento de placas de vídeo modernas e da API CUDA para treinar modelos capazes de sintetizar imagens fotorrealistas a partir de um conjunto de exemplares iniciais, já tendo sido utilizada para criar modelos que geram faces humanas praticamente indistinguíveis de fotografias verdadeiras [16].

1.1 Contexto

Sob a ótica da criatividade computacional, é inevitável enxergar as GANs como promissoras ferramentas para geração de conteúdo e apoio criativo [38] [37]. Dentro dessa área, diversas abordagens já foram utilizadas para extrair resultados artísticos a partir de tais redes, como por exemplo modificações na arquitetura para maximizar o distanciamento de estilos pré-concebidos [24].

Existe, porém, uma barreira prominente e pouco discutida nesse processo, que é a necessidade de conhecimento técnico para trabalhar com tais ferramentas, tornando-as inacessíveis pela vasta maioria da comunidade artística em si, limitando-as aos que têm também conhecimento no domínio da programação. Além disso, os requisitos de hardware (normalmente *GPUs* poderosas) e elevados tempos de treinamento para obtenção de bons modelos restringem ainda mais tal acesso.

Ainda que o usuário consiga realizar o treinamento de seu modelo ou mesmo opte pelo uso de uma instância já treinada por terceiros, ainda existe uma última barreira que acontece no

momento da geração de imagens. Nesta etapa, a StyleGan por padrão recebe como entrada um conjunto de *seeds* que serão utilizadas para sintetizar uma imagem a partir de um dado modelo. Tais *seeds*, porém, não são representativas das características do espaço latente em questão, não permitindo assim a exploração dos atributos presentes nas imagens produzidas e limitando o potencial da rede como agente de apoio à inspiração, uma vez que esta terá de se restringir a gerar imagens de maneira aleatória e sem relação com as produções anteriores.

1.2 Objetivos

Este trabalho tem como objetivo atacar os problemas anteriormente citados e propor soluções que possibilitem tornar as redes adversárias generativas em ferramentas de expressão artística mais acessíveis para indivíduos que não possuem vasto conhecimento de programação. Outro objetivo é o desenvolvimento de uma nova abordagem para permitir a exploração do espaço latente presente nestas redes com o intuito de auxiliar o artista a encontrar inspirações dentro de um determinado modelo.

Para isto, será utilizada como base a mais recente GAN lançada pela Nvidia, a StyleGan2-ADA [30] na sua implementação em PyTorch. Esta rede, além de ser mais rápida e eficiente em comparação com suas versões anteriores, conta com um novo mecanismo discriminador que a permite ser treinada com uma quantidade vastamente inferior de imagens iniciais sem a perda qualidade dos resultados ou geração de *overfitting*. Deste modo, é mais viável que um artista possa criar um conjunto de dados inicial utilizando suas obras anteriores e inspirações já conhecidas como base sem precisar se preocupar em conseguir um número inviável de exemplares.

CAPÍTULO 2

Redes Adversárias Generativas

Para compreender melhor o funcionamento das GANs, é necessário primeiro entender o domínio no qual elas estão inseridas, que seria a modelagem generativa. A ideia básica desse tipo de modelagem consiste em pegar um conjunto de dados inicial e criar uma função de distribuição probabilística capaz de descrever o grau de pertinência de um exemplar dentro dela. Por meio de técnicas de amostragem, é possível utilizar essa função para gerar novos exemplares que têm pertinência com relação aos originais [25].

Existem diversos casos de uso que justificam o uso de modelos generativos, sendo alguns exemplos:

- Simulação fácil e não-custosa de tarefas que seriam inviáveis de serem testadas na prática, como boa parte dos sistemas críticos.
- Previsão de cenários futuros, sendo capaz de testar várias ideias para atuar em situações diferentes e selecionar a melhor destas.
- Tratamento de *missing data*, podendo completar as lacunas que estão faltando.
- Ampliação de conjuntos de dados pequenos por meio da geração de novos exemplares.
- Solução de tarefas que requerem a síntese de novos exemplares realistas dentro de um domínio, como o que será realizado neste trabalho.

Visando criar uma nova forma mais eficaz para resolver problemas como estes, Ian Goodfellow e seus colegas de doutorado arquitetaram a estrutura da GAN em 2014 [26]. Sua inspiração mais direta foi o estudo de Michael Gutmann e Aapo Hyvärinen sobre estimativa de contraste de ruído [27], o qual estudaram a fundo durante o doutorado.

2.1 Funcionamento da GAN

Como mencionado anteriormente, em uma GAN temos dois agentes que são adversários um do outro. O primeiro destes é o Gerador, responsável por criar novos exemplares que sejam semelhantes aos encontrados no conjunto inicial. Seu oponente é o Discriminador, responsável por avaliar exemplares e classificar se pertencem ou não aos originais. Estes agentes competem ciclicamente em um jogo de soma-zero onde o Gerador aprende a produzir exemplares cada vez melhores para enganar o Discriminador, que por sua vez gradualmente aprende a distinguir com maior precisão os falsos dos originais.

Durante o treinamento, dentro do Gerador vai sendo criado e aprimorado um atributo chamado de espaço latente. Este espaço é uma representação utilizada pelas redes de aprendizado de máquina para agrupar dados com características similares próximos uns aos outros tratando-os como pontos em um espaço n-dimensional no qual cada dimensão (ou conjunto destas) pode representar características intrínsecas dos objetos reais utilizados em seu treinamento. Cada ponto neste espaço, por sua vez, pode ser mapeado para uma representação de um destes objetos, sendo assim a maneira com a qual são gerados novos exemplares.

Para exemplificar este conceito, é possível visualizar uma rede que gera imagens de frutas e um ponto x pertencente ao espaço latente desta que é mapeado para a imagem de uma maçã. Pontos próximos de x provavelmente serão mapeados para outras imagens de maçãs, sendo algumas praticamente idênticas caso não sejam suficientemente distantes. Distanciar-se em apenas umas dimensão pode fazer com que as imagens geradas sejam diferentes em apenas alguns detalhes mais simples, como a cor passar de vermelha para verde. Por outro lado, para encontrar um ponto y que gere uma banana, possivelmente seria necessário se distanciar bastante de x , pois suas representações não compartilham muitas características em comum. Ainda assim, traçando uma linha entre estes dois pontos e gerando exemplares utilizando pontos intermediários, seria possível ver a maçã gradualmente tornando-se uma banana.

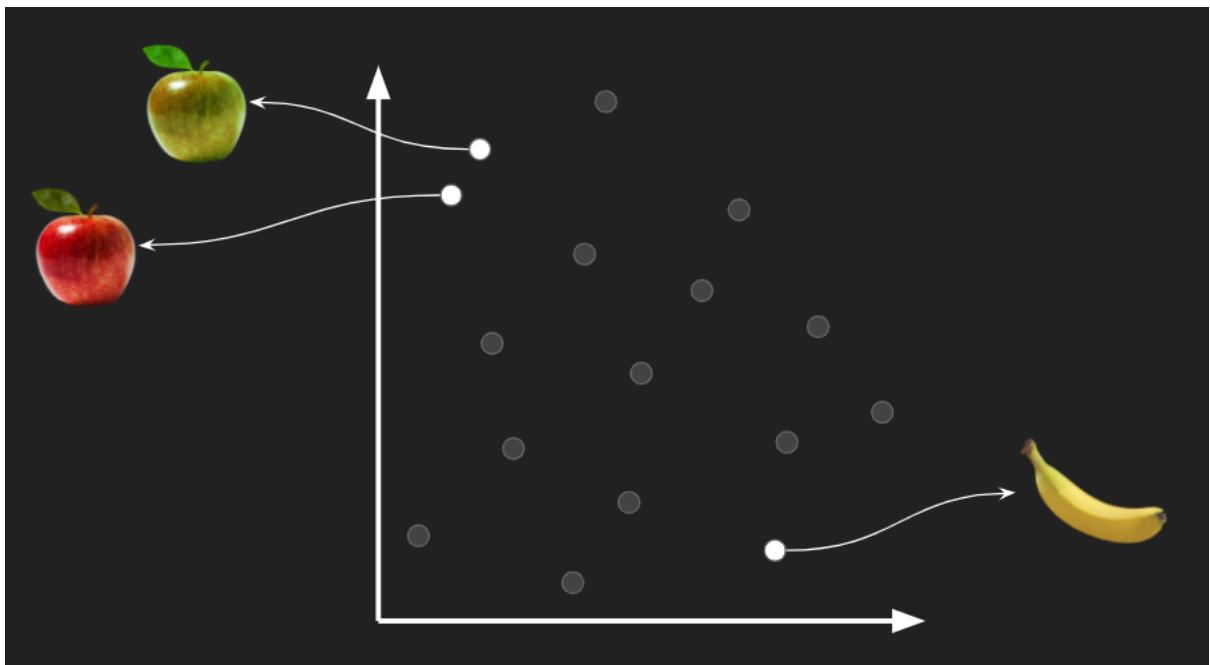


Figura 2.1: Exemplo de possíveis pontos em um espaço latente simplificado bidimensional e as imagens que estes poderiam gerar. Na prática, tais espaços normalmente serão constituídos por centenas de dimensões.



Figura 2.2: Interpolação de pontos em um modelo de faces humanas produzida por Jamshed Khan [32]. Existem inúmeras imagens intermediárias entre duas quaisquer no espaço latente.

Em comparação com as outras soluções na área de modelagem generativa, as GANs apresentam diversas vantagens que justificam seu uso, especialmente para a síntese de imagens. Primeiramente, elas utilizam um espaço latente no qual cada vetor representa um resultado de saída, permitindo o mapeamento destes vetores e a realização de operações aritméticas entre eles mantendo a coerência com relação às características que cada dimensão daquele espaço representa. Além disso, a síntese de exemplares é assintoticamente consistente (garantindo assim que a distribuição probabilística será eventualmente encontrada independentemente de qual seja o conjunto de dados inicial) e os resultados produzidos são constantemente classificados como os melhores, apesar de isto ser difícil de metrificar pois há bastante subjetividade em tal métrica.

Por tratar-se de um modelo teórico com bastante espaço para o uso de diferentes técnicas em cada uma de suas etapas, hoje já existem diversas implementações diferentes de GANs desenvolvidas por pessoas e organizações de todo o mundo. Uma vez que as GANs são comumente consideradas as melhores redes para a síntese de imagens [43], é neste domínio que concentra-se uma grande parte de tais implementações, como demonstrado na Figura 2.3.

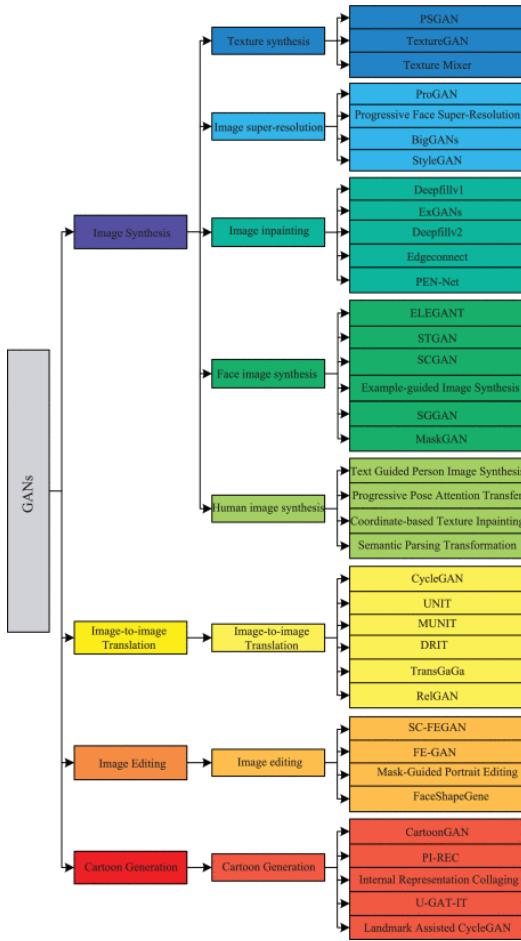


Figura 2.3: Diagrama taxonômico das GANs sintetizadores de imagem proposto por Chen et al. [43].

2.2 StyleGAN

A StyleGAN é uma implementação de GAN desenvolvida por pesquisadores da Nvidia e voltada para a sintetização de imagens que, desde seu lançamento em 2018, vem constantemente revolucionando a área por seus excelentes resultados.

Em 2019 a tecnologia ganhou grande atenção na mídia após ser utilizada por Phillip Wang para criar o site This Person Does Not Exist [16], que rapidamente viralizou na internet mostrando o poder deste tipo de rede. O site em questão utiliza um modelo treinado em mais de 70 mil fotos de faces humanas obtidas da base de dados do Flickr [2] para sintetizar imagens fotorrealistas de pessoas que não existem. O propósito por trás do projeto, segundo o autor, é trazer ao público geral uma conscientização sobre o atual poder de criação de conteúdo falso que pode facilmente se passar por verdadeiro, como por exemplo deepfakes [34].



Figura 2.4: Imagens de pessoas que não existem geradas pelo site This Person Does Not Exist.

O diferencial desta GAN consiste principalmente em reestruturar o Gerador para que este aprenda os “estilos” das imagens originais, como demonstrado na Figura 2.5. Estes “estilos” seriam características específicas daquele contexto e fariam com que as dimensões do espaço gerado melhor representassem tais características. Além disso, ela é capaz de separar atributos de alto nível e atributos estocásticos nas imagens (ou seja, distinguir entre aspectos mais “fixos” ou mais “aleatórios”), facilitando assim a detecção do que é mais importante para ser considerado durante o treinamento [31].

Em sua versão mais recente, a StyleGan2-ADA [30], agora também estão sendo realizadas modificações no Discriminador para que este permita um treinamento estável com números reduzidos de imagens originais. Por isso, essa versão se mostra como uma promissora candidata para o uso em projetos de menor escala onde existem limitações na quantidade de exemplares que podem ser obtidos.

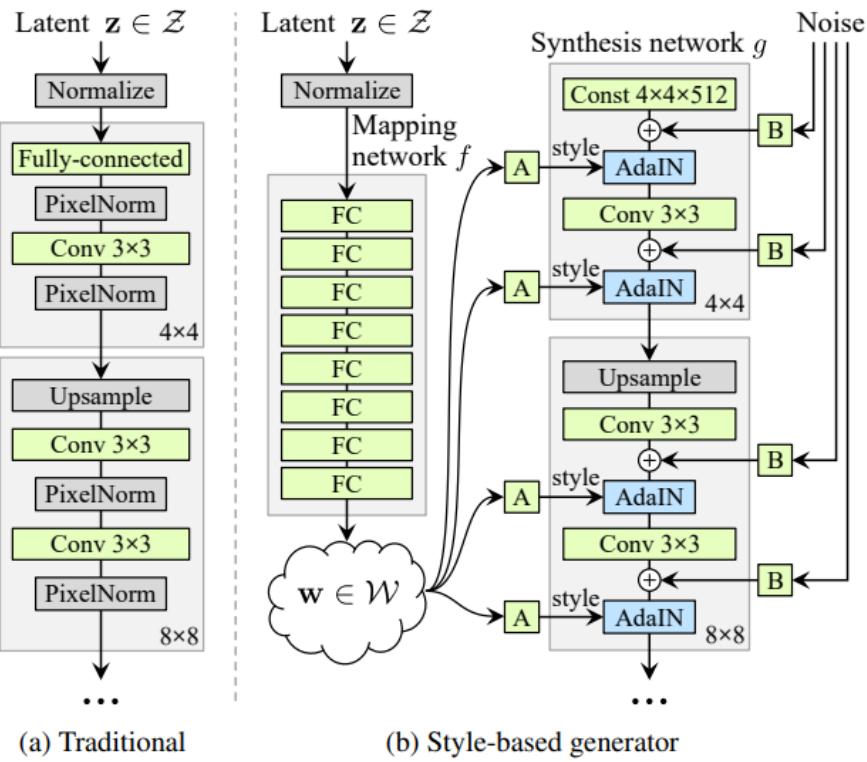


Figura 2.5: Comparaçāo entre a arquitetura de uma GAN padrāo e de uma StyleGAN, obtida do artigo original do projeto [31].

CAPÍTULO 3

Aplicações Artísticas das GANs

Apesar das barreiras técnicas existentes para utilizar essas tecnologias, existem hoje diversos artistas por todo o mundo que exploram cada vez mais como utilizar as redes adversárias geradoras como forma de expressão artística. A riqueza de diversidade encontrada nas maneiras distintivamente criativas com as quais GANs vêm sendo utilizadas nas obras já produzidas, por sua vez, corroboram fortemente a ideia de que estas são ferramentas que tem potencial para amplificar a visão artística do usuário, não substituindo-a.

Em geral, o surgimento das GANs teve um impacto gigantesco na comunidade já existente de artistas que utilizam inteligência artificial como ferramenta, abrindo cada vez mais caminho para seu reconhecimento dentro do mundo artístico convencional devido aos resultados alcançados. Dessa maneira, temos hoje arte criada utilizando estas redes sendo expostas em museus e até vendidas por elevados preços. O quesito de comercialização deste tipo de obras, que até então era relativamente duvidoso dada a possibilidade de replicação por serem criadas no meio digital, pode inclusive agora estar próximo de uma mudança de paradigmas devido à recente criação e popularização dos NFTs (*non-fungible tokens*).

No contexto da tecnologia em si, já foram realizados também trabalhos para adaptar o modelo da GAN com o objetivo de adicionar um teor artístico em seus resultados sintetizados. Um exemplar em destaque nesse contexto seriam as Redes Criativas Adversárias (CANs) [24], que por meio de modificações na etapa de treinamento da rede, conseguem se distanciar de estilos pré-concebidos enquanto ainda se mantêm próximas da distribuição artística conhecida.

3.1 Exemplos de Obras

A seguir serão apresentados alguns exemplares do que já foi produzido por artistas utilizando as redes adversárias geradoras.

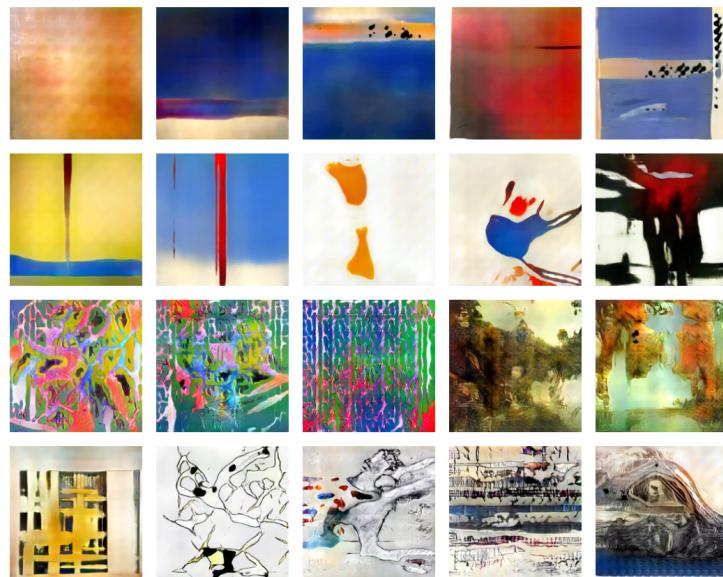


Figura 3.1: Coleção de 2018 da AICAN [1], que utiliza CANs [24] para gerar obras que a maior parte das pessoas não consegue distinguir que foram criadas por um computador. As obras em questão foram expostas em museus nos Estados Unidos e Europa.



Figura 3.2: Obra do Ganvas Studio [5] exposta em parede. Este estúdio foca-se em trazer à vida obras geradas por máquinas utilizando técnicas de impressão capazes de captar a essência e nuances de cada exemplar.

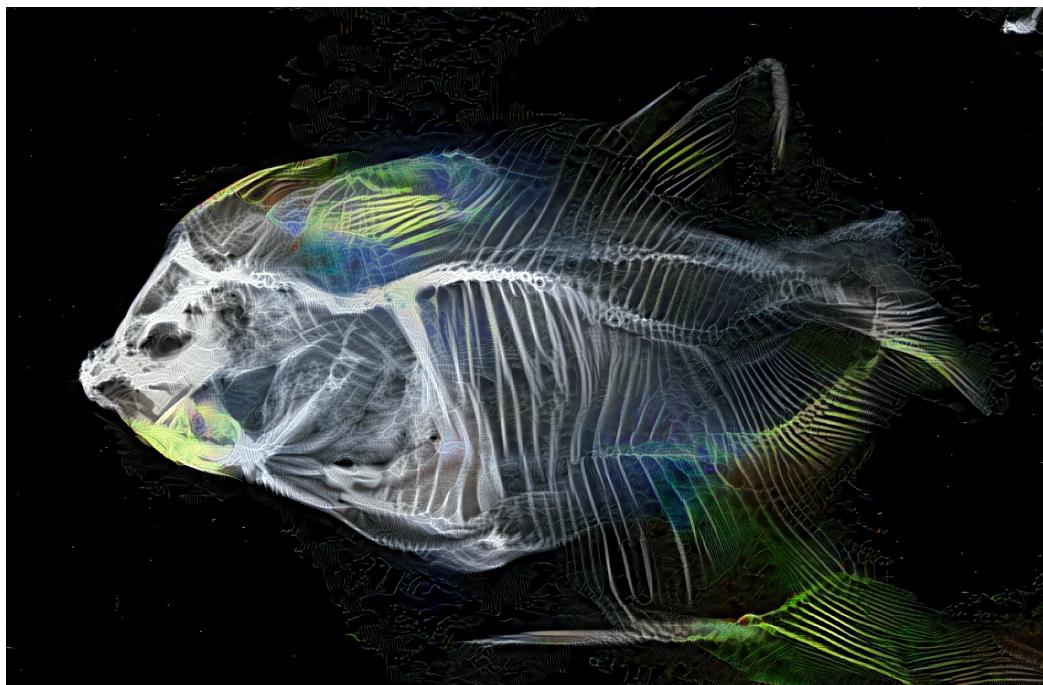


Figura 3.3: Obra do Neural Zoo de Sofia Crespo [23], que utiliza simultaneamente GANs e redes neurais convolucionais para criar criaturas e componentes naturais fantásticos.



Figura 3.4: Quadros da produção *Learning to See* de Memo Akten [18], na qual o artista usa objetos do dia-a-dia e partes do corpo como entrada para uma GAN que as transforma em elementos naturais. Este processo é aplicado em quadros de um vídeo para gerar uma saída fluída também em forma de vídeo.

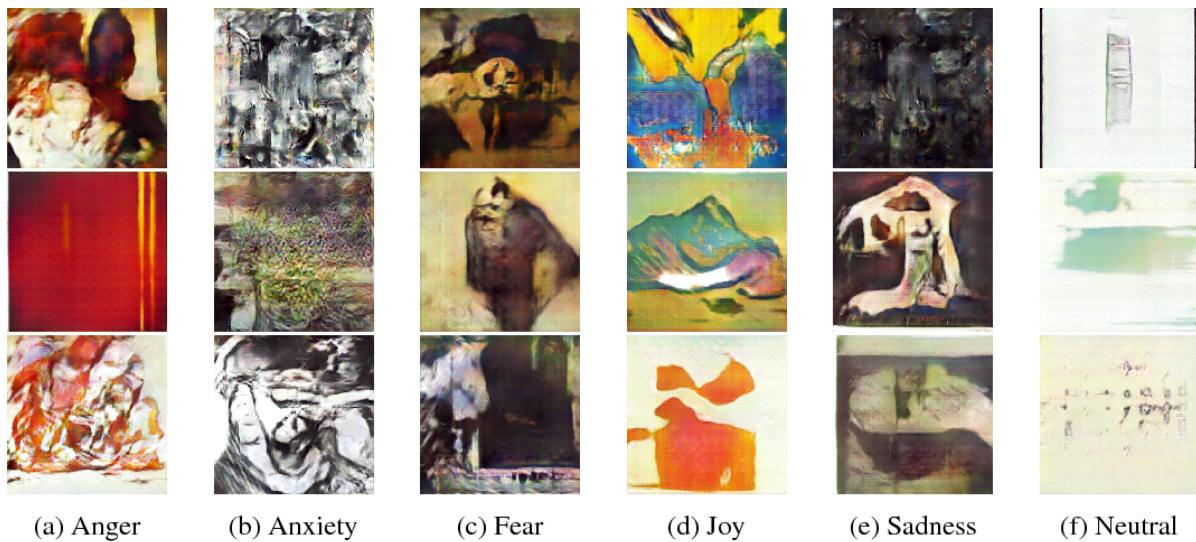


Figura 3.5: Exemplares gerados pela Emotional GAN [19] utilizando análise de sentimento como entrada.



Figura 3.6: Obras de Helena Sarin [9], que experimenta com diferentes implementações de GANs para cada uma de suas produções.



Figura 3.7: Unknown Departure [42], uma obra criada combinando duas GANs. A primeira foi treinada com imagens de rodovias enquanto a segunda captou o estilo de um desenho feito com carvão.

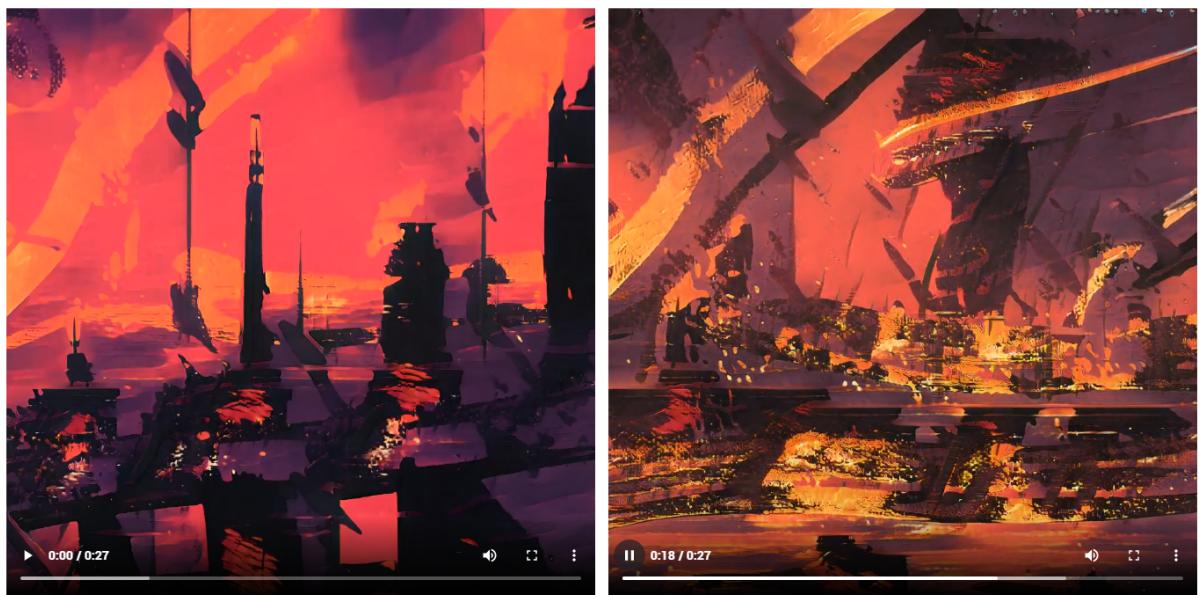


Figura 3.8: Dois momentos distintos de uma exploração audiovisual do espaço latente de uma StyleGan [22].



Figura 3.9: Suéter produzido pela HATRA (ハトラ) a partir de uma GAN treinada com imagens de pássaros [29]. Esta peça faz parte da coleção Synthetic Feather, toda criada com essa tecnologia.



Figura 3.10: Exemplares gerados pelo Toonify Yourself [36], uma rede StyleGan que utilizou *transfer learning* a partir de imagens de faces humanas para faces de personagens de desenhos animados do cinema.



Figura 3.11: Processo de treinamento de uma GAN realizado por Yuguang Zhang ao gravar a si mesmo dormindo para captar o impacto de suas posições no lençol [44]. O resultado foi posteriormente utilizado para produzir um vídeo simulando uma interação semi-humana semi-objeto em sua cama.

CAPÍTULO 4

Preparação do Conjunto de Dados

Para obter um modelo utilizável de uma StyleGAN, é necessário primeiramente passar por dois passos básicos. O primeiro destes seria a preparação e pré-processamento de um conjunto de dados que possa ser interpretado como entrada para a rede. Em seguida, é indispensável a realização das iterações de treinamento para que o modelo aprenda o estilo das imagens originais e se torne capaz de produzir novos exemplares.

Desta maneira, a primeira etapa básica para a construção de uma rede StyleGan torna-se, inevitavelmente, a escolha das imagens que serão utilizadas como conjunto de dados inicial, sob as quais a rede aprenderá as características desejadas para os exemplares futuramente gerados. É importante que os exemplares utilizados nesta etapa sejam em sua maioria de alguma maneira representativos do tipo de imagem cuja síntese é desejada a partir do modelo após seu treinamento. Estes também devem, idealmente, ter alguma relação estética entre si, caso contrário o modelo pode não conseguir convergir em um resultado de forma coerente.

Neste momento, é essencial também ter cuidado para não escolher imagens que possam fazer a rede convergir com *overfitting*, um dos problemas mais comuns para a maior parte das arquiteturas de aprendizado de máquina. Este termo é utilizado para descrever uma rede cujos resultados sintetizados são muito parecidos ou praticamente idênticos ao conjunto de dados inicial, perdendo assim o seu valor. No caso da StyleGan, isso pode ocorrer quando o conjunto de imagens iniciais é muito pequeno ou quando são muito semelhantes umas às outras

4.1 Seleção de Imagens

Devido às características já citadas da StyleGan2-ADA, nela são necessários menos exemplares iniciais para produzir um modelo sem *overfitting* em comparação a outras redes similares, sendo cerca de 1000 exemplos um bom ponto de partida para resultados interessantes baseado em algumas aplicações já testadas anteriormente [30]. No contexto artístico, porém, como o objetivo não é a perfeição e fotorrealismo, é possível que menos exemplares já sejam suficientes para a realização de um treinamento frutífero, sendo necessários testes futuros para determinar valores recomendados. Evidentemente, quanto mais imagens diferentes forem utilizadas, maiores serão a variação e diversidade encontradas no espaço latente.

É importante ressaltar também que é possível treinar um modelo utilizando outro previamente treinado como base por meio da técnica de *transfer learning*. Uma vez que este recurso é normalmente utilizado para obter um tempo de treinamento reduzido e resultados melhores, costuma-se desejar que o modelo do qual o aprendizado será transferido seja de um domínio similar ao das imagens com as quais será realizado o novo treinamento, pois assim é possível o

aproveitamento de várias características do espaço latente já existente. Empiricamente, porém, é possível perceber que esta não é necessariamente a melhor abordagem no processo de criação de um modelo para fins artísticos.

Uma vez que na arte não se busca normalmente a perfeição absoluta, mas sim novos conceitos que por muitas vezes surgem da mistura de outros já existentes, realizar *transfer learning* a partir de um modelo em um domínio distante dos exemplares que serão utilizados para o treinamento tem potencial para gerar excelentes resultados. A mesma lógica se aplica na montagem do conjunto de dados inicial, no qual podem ser inseridas imagens que não possuem muitas similaridades umas com as outras, mas cujas características acabarão influenciando a formação do espaço latente para construir novas possibilidades que não seriam possíveis sem estas. Alguns exemplos nesse sentido seriam imagens com cores, formas e texturas diferentes das encontradas nas outras, que trariam mais diversidade à rede além de servirem para aumentar a quantidade de exemplos e reduzir o risco de *overfitting*.



Figura 4.1: Imagens aleatórias geradas por um modelo treinado com exemplares do livro *Kunstformen der Natur* [28], tendo usado como base um *transfer learning* a partir de um modelo de faces humanas. É possível observar a presença de algumas características do modelo anterior ainda presente neste, como olhos, bocas e óculos, além do formato geral de um rosto. Tais características tendem a eventualmente desaparecer à medida que o treinamento durar mais tempo.

4.2 Pré-Processamento

O conjunto de dados utilizado para o treinamento de uma StyleGan, porém, não pode ser composto simplesmente de várias imagens arbitrárias, tendo estas que seguir um padrão simples: todas devem ser quadradas e ter como tamanho de seus lados um valor que seja potência de 2, além de pertencerem ao mesmo espaço de cores. Apesar de parecer superficialmente simples, este requisito faz com que a maior parte dos conjuntos de dados necessite passar por um pré-processamento que seria temporalmente custoso para um usuário que não domine o uso de ferramentas de automação. Além disso, por fim, ainda é necessário executar um script do próprio StyleGan2-ADA para que tais imagens sejam convertidas para o formato aceito pela rede, elevando ainda mais a complexidade técnica desta etapa.

Visando tornar este pré-processamento acessível para todo tipo de usuário, produzimos um Jupyter Notebook [10] para ser executado na plataforma Google Colaboratory [6] que é capaz de fazer todo este processo de maneira intuitiva e interativa, lendo e escrevendo as imagens por meio do Google Drive [7] do usuário.

A plataforma em questão foi escolhida por permitir a criação gratuita de um ambiente de execução em nuvem para Jupyter Notebooks, contando com acesso a placas de vídeo da Nvidia que, por sua vez, são necessárias para a realização deste tipo de projeto. Além disso, a natureza dos *notebooks* permite a intercalação de código e texto, tornando assim possível guiar o usuário durante seu uso e requisitando dele apenas a edição dos valores de algumas variáveis-parâmetros e a execução das células sequencialmente, ações simples que podem ser facilmente introduzidas.

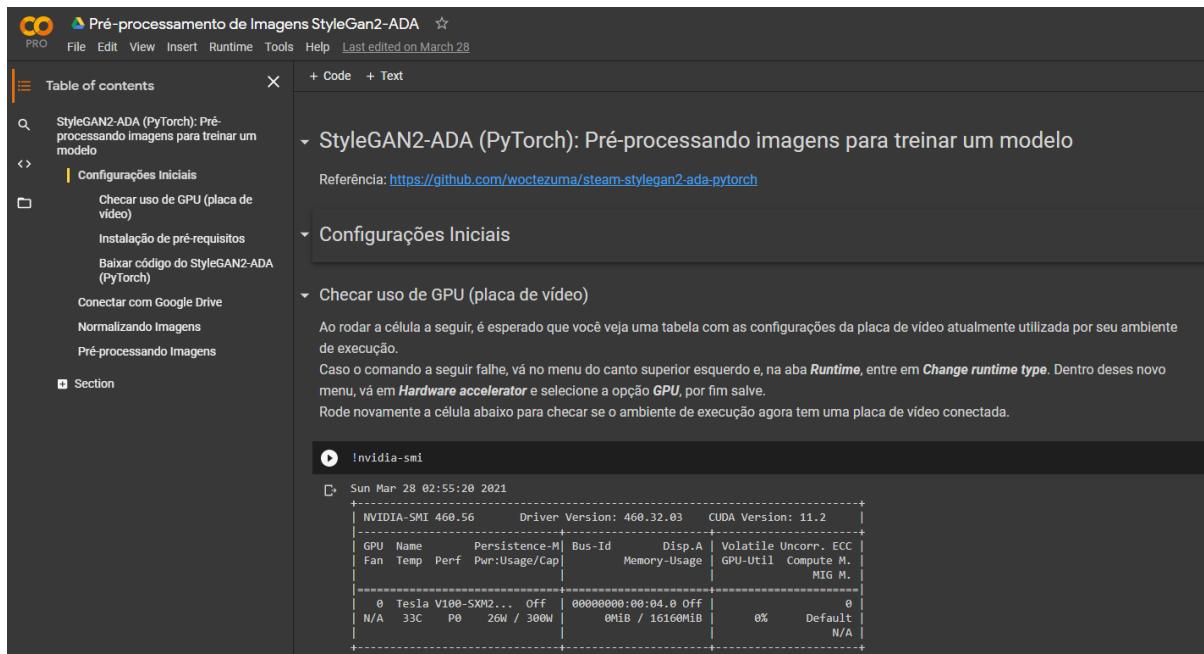


Figura 4.2: Visão geral da plataforma Google Colaboratory. As células textuais permitem guiar o usuário por meio de comandos simples enquanto as células de código deverão ser executadas a partir destas instruções.

O *notebook* inicia com uma série de configurações iniciais simples como a conexão do ambiente de execução com uma GPU, download de pré-requisitos e conexão com o Google Drive. Logo em seguida é requisitado do usuário o tamanho desejado para as imagens geradas, assim como a heurística a ser utilizada para ajustá-las em formato quadrado. Para realizar esta transformação, é utilizada a biblioteca OpenCV [20], sendo esta capaz de expandir de diversas maneiras as bordas da menor dimensão da imagem até que esta tenha o mesmo valor de altura e largura. Por fim é executado um *script* que processa todas as imagens e salva suas versões modificadas em uma nova pasta no Google Drive.

A última célula deste *notebook* completa a etapa de pré-processamento lendo os resultados do passo anterior e executando o comando da StyleGan2-ADA que transforma as imagens geradas em um conjunto de dados capaz de ser interpretado pela rede na futura etapa de treinamento. Com isto, toda a execução do código necessário para este processo torna-se uma espécie de “caixa-preta” na qual o conhecimento técnico sobre a implementação é abstraído do usuário.



(a) Imagem original de um violoncelo da Filarmônica de Paris anterior à etapa de pré-processamento.



(b) Imagem de um violoncelo da Filarmônica de Paris após a etapa de pré-processamento.

Figura 4.3: Comparação entre imagens antes e depois da etapa de pré-processamento.

O *notebook* utilizado nesta etapa pode ser encontrado em <https://colab.research.google.com/drive/1DDAwNp1CwRt9aa0XNACzyXCw8DV-xUg3?usp=sharing>.

Para realizar o pré-processamento com imagens próprias, basta acessar o *link* acima e seguir as instruções na interface do Colaboratory.

CAPÍTULO 5

Treinamento do Modelo

Tendo em mãos o conjunto de dados já processado com as imagens a serem utilizadas, a realização do treinamento de uma StyleGAN2-ADA torna-se uma tarefa simples, sendo os requisitos de hardware a única barreira nesta etapa. Para usuários mais experientes, existe a possibilidade de experimentar com alguns parâmetros para esta etapa como *gamma* e *mini_batch*, porém os valores padrão para estes são suficientemente aceitáveis para uso geral.

Este também é o momento no qual deverá ser realizada a escolha em relação ao possível uso de *transfer learning*. Caso seja optado pelo uso deste recurso, a API da StyleGAN2-ADA permite o uso de qualquer modelo pré-treinado que o usuário possua, inclusive tendo este sido treinado em versões anteriores do StyleGAN. Além disso, é oferecida também uma seleção de alguns modelos padrão que já vem acoplados na atual implementação da rede.

5.1 Execução

Para esta etapa foi criado mais um *notebook* para ser utilizado com os recursos fornecidos pelo Colaboratory. Lendo os arquivos no Google Drive do usuário, este *notebook* consegue partir de onde o anterior terminou e utilizar os dados para alimentar o estado inicial do treinamento.

É importante ressaltar que o tempo necessário para o treinamento de uma rede como estas é bastante dependente do tamanho do conjunto de dados utilizado, além das dimensões das imagens ali contidas. Em média, pode-se esperar que o processo leve ao menos alguns dias para que consiga convergir em um resultado estável. Como o ambiente de execução do Google Colaboratory conta com algumas limitações em relação ao tempo máximo de execução contínua (em geral menos de 24 horas), se faz necessário um mecanismo capaz de salvar versões intermediárias do modelo regularmente e retomar o treinamento a partir da mais recente destas.

Para possibilitar a divisão do treinamento em etapas capazes de serem processadas na plataforma em questão, foi utilizada como base para a implementação presente no *notebook* uma *fork* do repositório do StyleGAN2-ADA [15] que permite aproveitar o sistema de *transfer learning* para retomar o treinamento a partir de uma etapa anterior.

Durante sua execução, o *notebook* salvará estados intermediários do modelo no Google Drive do usuário e estes poderão ser referenciados como pontos de retomada na próxima execução. Para cada modelo salvo, é gerada também uma imagem contendo um conjunto de 120 exemplares sintetizados a partir dele. Com isto, o usuário pode averiguar os resultados atuais e optar por terminar o treinamento em qualquer ponto durante sua execução caso esteja satisfeito com os resultados.



Figura 5.1: Exemplo de um conjunto de imagens geradas a partir de um modelo intermediário.

tick 0	kimg 0.0	time 50s	sec/tick 6.3	sec/kimg 197.43	maintenance 43.9	cpumem 4.19	gpumem 12.39	augment 0.000
tick 1	King 4.0	time 3m 33s	sec/tick 156.1	sec/kimg 39.02	maintenance 6.3	cpumem 4.62	gpumem 9.47	augment 0.004
tick 2	kimg 8.0	time 6m 09s	sec/tick 156.6	sec/kimg 39.15	maintenance 0.1	cpumem 4.62	gpumem 9.47	augment 0.008
tick 3	king 12.0	time 8m 46s	sec/tick 156.8	sec/kimg 39.20	maintenance 0.1	cpumem 4.62	gpumem 9.48	augment 0.013
tick 4	King 16.0	time 11m 23s	sec/tick 156.8	sec/kimg 39.20	maintenance 0.0	cpumem 4.62	gpumem 9.54	augment 0.018
tick 5	kimg 20.0	time 14m 00s	sec/tick 156.9	sec/kimg 39.21	maintenance 0.1	cpumem 4.62	gpumem 9.51	augment 0.022
tick 6	king 24.0	time 16m 42s	sec/tick 156.5	sec/kimg 39.11	maintenance 5.7	cpumem 4.92	gpumem 9.52	augment 0.026
tick 7	King 28.0	time 19m 19s	sec/tick 157.1	sec/kimg 39.27	maintenance 0.1	cpumem 4.92	gpumem 9.50	augment 0.030
tick 8	kimg 32.0	time 21m 56s	sec/tick 157.0	sec/kimg 39.25	maintenance 0.0	cpumem 4.92	gpumem 9.53	augment 0.035
tick 9	king 36.0	time 24m 33s	sec/tick 157.1	sec/kimg 39.27	maintenance 0.1	cpumem 4.92	gpumem 9.56	augment 0.042
tick 10	King 40.0	time 27m 11s	sec/tick 157.1	sec/kimg 39.28	maintenance 0.1	cpumem 4.92	gpumem 9.55	augment 0.048
tick 11	kimg 44.0	time 29m 54s	sec/tick 156.8	sec/kimg 39.20	maintenance 6.0	cpumem 4.92	gpumem 9.54	augment 0.055
tick 12	king 48.0	time 32m 31s	sec/tick 157.2	sec/kimg 39.38	maintenance 0.0	cpumem 4.92	gpumem 9.62	augment 0.062
tick 13	King 52.0	time 35m 08s	sec/tick 157.3	sec/kimg 39.32	maintenance 0.1	cpumem 4.92	gpumem 9.61	augment 0.069
tick 14	kimg 56.0	time 37m 46s	sec/tick 157.3	sec/kimg 39.33	maintenance 0.1	cpumem 4.92	gpumem 9.61	augment 0.076
tick 15	king 60.0	time 40m 23s	sec/tick 157.4	sec/kimg 39.36	maintenance 0.1	cpumem 4.92	gpumem 9.59	augment 0.084
tick 16	King 64.0	time 43m 06s	sec/tick 157.1	sec/kimg 39.27	maintenance 5.8	cpumem 4.99	gpumem 9.64	augment 0.092
tick 17	kimg 68.0	time 45m 43s	sec/tick 156.9	sec/kimg 39.23	maintenance 0.4	cpumem 4.99	gpumem 9.58	augment 0.100
tick 18	King 72.0	time 48m 21s	sec/tick 157.4	sec/kimg 39.36	maintenance 0.1	cpumem 4.99	gpumem 9.64	augment 0.107
tick 19	kimg 76.0	time 50m 59s	sec/tick 157.6	sec/kimg 39.40	maintenance 0.1	cpumem 4.99	gpumem 9.58	augment 0.114
tick 20	king 80.0	time 53m 36s	sec/tick 157.5	sec/kimg 39.37	maintenance 0.0	cpumem 4.99	gpumem 9.68	augment 0.122

Figura 5.2: *Logs* de execução de um treinamento. Cada *tick* é capaz de gerar uma *snapshot* a partir do qual o treinamento pode ser retomado futuramente após ser interrompido. Para não gerar um número maior que o necessário de arquivos intermediários, é possível ajustar o parâmetro *snap*, que representa o intervalo de *ticks* entre uma geração e outra.

O *notebook* utilizado nesta etapa pode ser encontrado em https://colab.research.google.com/drive/1sFSq-JZPgkSrlrGNVvnLyHI_7GWgj8VU?usp=sharing.

Para realizar o treinamento a partir de um conjunto de imagens já processadas resultantes da etapa anterior, basta acessar o *link* acima e seguir as instruções na interface do Colaboratory.

CAPÍTULO 6

Exploração do Espaço Latente

Como já visto anteriormente, o espaço latente da GAN é constituído por diversas dimensões que representam as características intrínsecas dos objetos reais com os quais a rede foi treinada. Dentro deste, cada ponto pode ser utilizado para sintetizar um novo exemplar dentro do domínio em questão. Pontos próximos sintetizarão exemplares similares, enquanto pontos distantes sintetizarão exemplares bastante distintos.

A exploração guiada de tal espaço em uma GAN é a chave para seu uso como agente de apoio à inspiração para artistas, uma vez que estes seriam capazes de direcionar o tipo de exemplares sintetizados a partir de seus gostos e visões artísticas. Sem uma maneira estruturada para caminhar por esse espaço constituído normalmente por centenas de dimensões, este processo teria que limitar-se à geração de imagens aleatórias sem qualquer agência do usuário. Isto então torna-se um problema de busca no qual o estado final desejado não é algo exato, mas sim um conceito subjetivo na mente do usuário, sendo necessária uma estratégia que leve em consideração tal subjetividade para resolvê-lo.

6.1 Metodologia

A rede utilizada neste trabalho, StyleGan2-ADA, conta com um espaço latente constituído por 512 dimensões, o que torna a exploração deste por uma usuário humano uma tarefa não-trivial uma vez que não há uma maneira direta para visualizar intuitivamente a representação do espaço como um todo. Cada ponto neste espaço, porém, pode ser facilmente visualizado pelo usuário ao ser mapeado para sua representação em forma de imagem. Com isto, é possível utilizar a percepção do usuário sobre estas imagens como entrada para direcionar os caminhos a serem explorados no espaço.

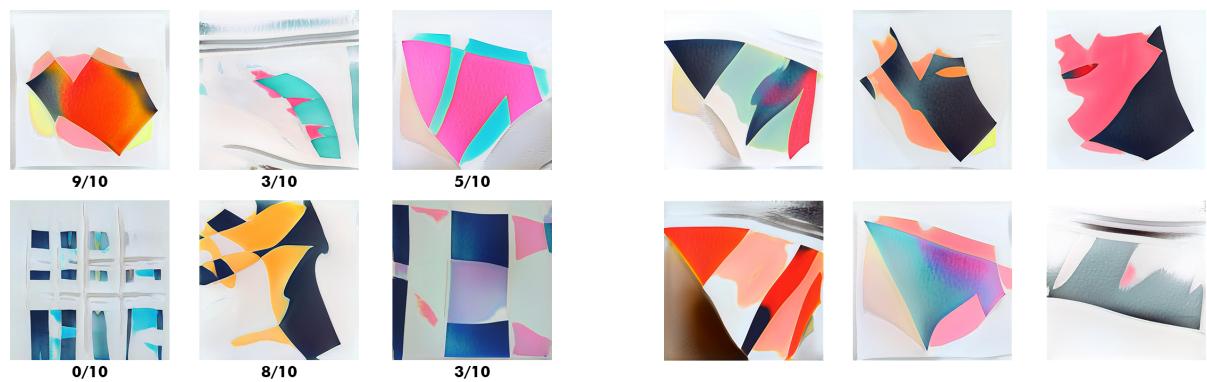
Para isto, foi definida uma estratégia de busca baseada em um sistema evolutivo na qual cada indivíduo da população tem como genótipo um vetor de 512 posições, cada uma representando uma dimensão do espaço latente. Estes indivíduos, por sua vez, podem ser representados por imagens ao utilizarmos seus genes como entrada para a função gerativa da rede em questão, demonstrando assim seus fenótipos. Todo este processo ocorrerá dentro de um algoritmo genético interativo no qual a função de *fitness* é controlada pela percepção do usuário, tendo sua implementação sido inspirada pelo livro *The Nature of Code* de Daniel Shiffman [39], que por sua vez baseou-se na metodologia utilizada pela exposição Galápagos de Karl Sims [40] [41].

Ao observar a população presente em cada geração do algoritmo, é solicitado do usuário que indique o quanto cada indivíduo (representado por uma imagem) está sendo útil como

inspiração para a visão artística que está seguindo naquele momento. Uma vez coletada esta entrada do usuário, o algoritmo a utiliza como função de *fitness* para evoluir a população ali presente por meio de um processo de acasalamento, recombinação e mutação. Após cada iteração, é de se esperar que as imagens sintetizadas vão cada vez mais convergir para uma direção capaz de inspirar ainda mais o usuário.

Nativamente, a StyleGan2-ADA não permite o uso direto de vetores como entrada para sintetização de imagens, possibilitando apenas o uso de *seeds* para gerá-los a partir de valores aleatórios. Uma vez que estas *seeds* não apresentam qualquer ligação direta entre genótipo e fenótipo, não seria possível aplicar nelas as técnicas de recombinação necessárias para o funcionamento do algoritmo genético. Por isso, foram criados novos métodos na *fork* utilizada do projeto para permitir a geração de imagens recebendo vetores como entrada.

Para permitir que o usuário de fato interaja com o algoritmo genético e o utilize como ferramenta de apoio à inspiração, foram implementadas duas interfaces distintas. A primeira consiste em uma aplicação *desktop*, enquanto a segunda foi desenvolvida de modo a ser utilizada diretamente pela plataforma do Google Colaboratory.



(a) População inicial gerada pelo algoritmo. Abaixo de cada indivíduo, encontra-se a nota atribuída a ele que será usada como função de *fitness*.

(b) População após uma iteração evolutiva do algoritmo genético.

Figura 6.1: Evolução de uma população utilizando o algoritmo genético em uma rede treinada com obras de Frea Buckler [4] [3]. Comparando as imagens, é possível perceber que os genes relacionados às cores e formas das imagens mais bem avaliadas da primeira geração tornaram-se características dominantes na segunda geração.

6.2 Aplicação Desktop

A primeira implementação de uma interface para a ferramenta em questão foi feita utilizando a biblioteca Tkinter [17] para gerar uma aplicação desktop. Após experimentações empíricas com componentes e valores diferentes, foi decidido que um componente *slider* simples com escala de 0 a 10 abaixo de cada imagem seria uma maneira intuitiva para o usuário de expressar seus gostos de maneira quantitativa, podendo esta ser traduzida facilmente para a função de *fitness*. Uma vez satisfeito com suas notas, o usuário pode utilizar o botão *update* para evoluir a

população para a próxima geração. Uma caixa de texto também foi colocada na interface para permitir a edição da taxa de mutação do sistema.

A aplicação desktop, porém, necessita que o usuário possua uma placa de vídeo da Nvidia em seu computador e instale os *drivers* do CUDA, assim como seu kit de desenvolvimento. Sendo assim, apesar de servir como prova de conceito, esta versão deixa a desejar no quesito de acessibilidade tanto por quesitos de hardware quanto de conhecimento técnico necessário.



Figura 6.2: Interface da aplicação desktop com exemplares gerados a partir de uma rede treinada com desenhos de figuras humanas [12].

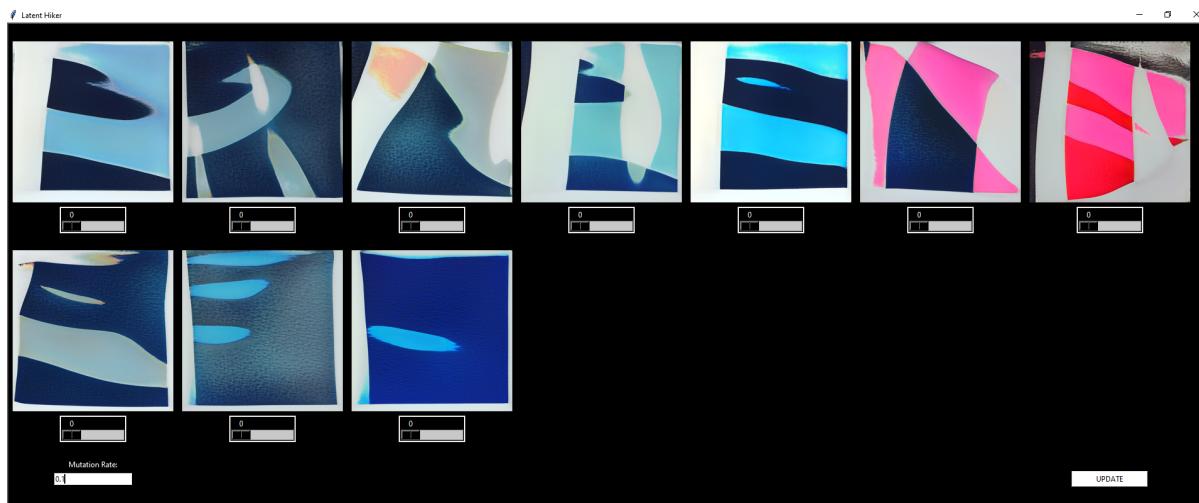


Figura 6.3: Interface da aplicação desktop com exemplares gerados a partir de uma rede treinada com obras de Frea Buckler.

A fork da StyleGAN2-ADA-Pytorch contemplando a implementação do algoritmo genético

e da aplicação desktop pode ser encontrada em <https://github.com/ClaudioCarvalhoo/stylegan2-ada-pytorch-latent-space-exploration>.

6.3 Notebook para Google Colaboratory

Visando resolver o problema de acessibilidade apresentado na implementação anterior, foi realizada uma tentativa de reproduzir a mesma interface dentro de um Jupyter Notebook com o intuito de executá-lo no Google Colaboratory, ambiente utilizado também nas etapas anteriores deste trabalho. Por meio dos Jupyter Widgets [11], esta conversão foi não apenas fácil, mas também mostrou-se mais simples de ser desenvolvida e atingiu resultados praticamente idênticos em termos de usabilidade.

Com isso, o processo completo desde o processamento das imagens, ao treinamento de uma rede, até a exploração do espaço latente, provou-se possível de ser realizado inteiramente na nuvem utilizando uma única plataforma centralizada com o apoio do Google Drive para armazenar os arquivos. Desta maneira, os requisitos de hardware proprietários por conta do usuário foram completamente eliminados.

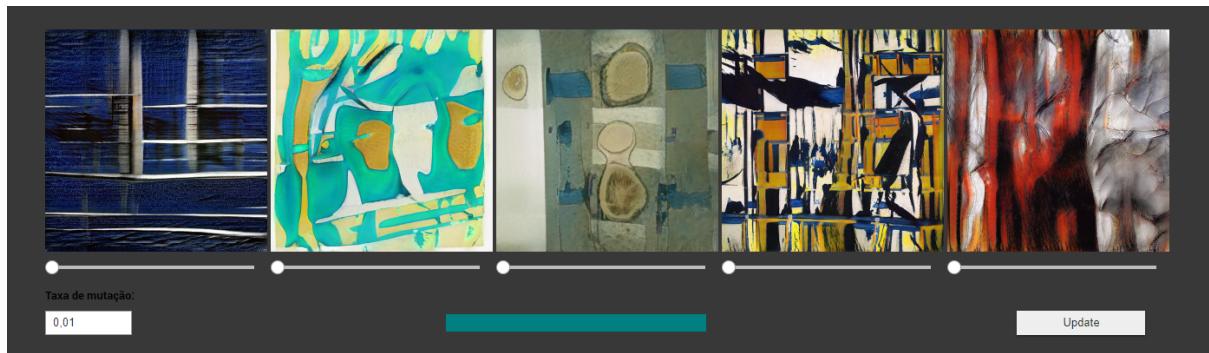


Figura 6.4: Interface executando em um *notebook* no Google Colaboratory com exemplares gerados a partir de uma rede treinada com obras de arte moderna [14].

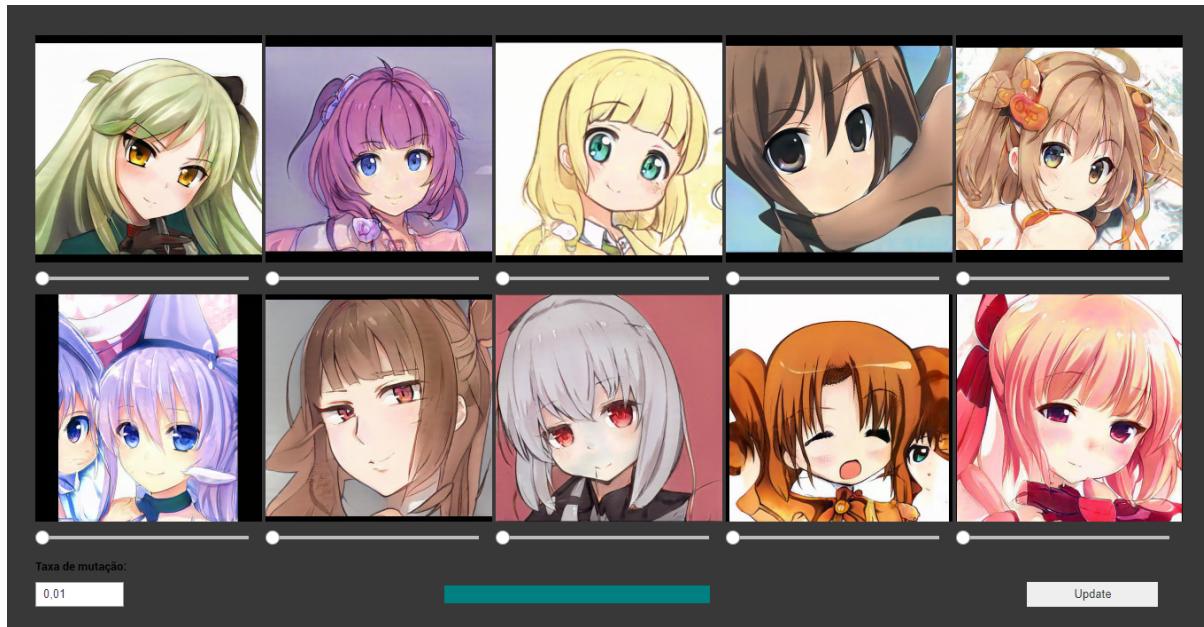


Figura 6.5: Interface executando em um *notebook* no Google Colaboratory com exemplares gerados a partir de uma rede treinada com faces de personagens de anime [13].

O *notebook* utilizado nesta implementação pode ser encontrado em <https://colab.research.google.com/drive/16YgKNrUawMoXt81wtmFc-efdD7LUM01Y?usp=sharing>.

Para realizar a exploração de um modelo de StyleGAN (podendo este ser treinado utilizando as etapas anteriores), basta acessar o *link* acima e seguir as instruções na interface do Colaboratory.

CAPÍTULO 7

Avaliação

Com o intuito de diminuir as barreiras técnicas para artistas, foi avaliada a experiência de uso das soluções desenvolvidas, baseando-se no nível de dificuldade percebido por usuários. Durante os testes, foi requisitado aos participantes que utilizassem as ferramentas desenvolvidas para realizar as três etapas propostas para o uso da GAN, sendo estas a preparação do conjunto de dados, treinamento da rede e exploração do espaço latente. Ao fim foram aplicados alguns questionários e realizadas entrevistas com o intuito de metrificar os resultados obtidos.

7.1 Metodologia

Todos os experimentos foram conduzidos de maneira remota utilizando a plataforma Google Meet [8] devido às limitações impostas pelo momento de pandemia da COVID-19. Também por este motivo, todos os usuários utilizaram a interface desenvolvida no Google Colaboratory para a etapa de exploração do espaço latente, uma vez que necessitariam de uma placa de vídeo dedicada em seus computadores para utilizar a aplicação desktop. Para garantir a consistência dos resultados, o roteiro utilizado para conduzir os testes também manteve-se constante.

Para cada usuário, inicialmente foi apresentada uma breve introdução sobre o objetivo do trabalho, apresentando alguns exemplares já existentes do uso artístico da StyleGan para servir de contexto e motivação. Em seguida, foi concedido acesso a uma conta no Google Drive na qual já se havia sido preparado um diretório contendo imagens de obras de Frea Buckler [3] que seriam utilizadas para realizar o treinamento. Esta conta foi importante para que não houvesse a necessidade de que cada um trouxesse suas próprias imagens, agilizando assim o processo.

Antes de começar o uso dos *notebooks*, foi realizada também uma explicação sobre a interface do Google Colaboratory, demonstrando as diferenças entre as células de texto e de código, além de esclarecer que eles deveriam apenas seguir o fluxo sequencialmente, interagindo com os botões e caixas de texto quando requisitados. Após isto, o uso de cada um foi observado por meio de compartilhamento de tela, sendo qualquer interferência evitada a não ser que usuário se encontrasse incapaz de progredir.

Como a etapa de treinamento tem duração esperada de alguns dias, foi acompanhado o uso até a célula em questão ser executada e, logo depois, os usuários foram instruídos a interromper manualmente a execução do *notebook* para testar o sistema de retomada da execução a partir de um *checkpoint*, simulando assim o comportamento do Colaboratory após algumas horas. Na conta do Google Drive fornecida foram disponibilizados também alguns modelos de redes já treinadas para serem utilizadas na etapa de exploração, simulando que o treinamento já houvesse sido concluído.

Durante os ciclos de exploração do espaço latente no último *notebook*, foi encorajado aos usuários que usassem o modelo que mais se adequasse ao seu estilo artístico e realizassem o número de iterações que desejassem, parando por conta própria quando achassem que havia sido suficiente. Por fim, foram entregues alguns formulários para que respondessem questões sobre a usabilidade da ferramenta como um todo, além de uma discussão breve para obter *feedbacks* de mais alto nível sobre o que acharam dos resultados.

Para mensurar os resultados obtidos nos testes, foram utilizadas duas escalas psicométricas: uma voltada para usabilidade e a outra para experiência de usuário. A primeira delas foi a System Usability Scale (SUS) [21], uma escala simples e rápida de se responder que é amplamente utilizada na indústria para obter uma média simples da usabilidade de sistemas. A outra foi o User Experience Questionnaire (UEQ) [33] que, por meio de um número maior de perguntas, consegue obter uma análise mais detalhada da experiência, dividindo seus resultados em seis métricas diferentes, sendo estas atratividade, clareza, eficiência, confiabilidade, estimulação e inovação.

As escalas utilizadas, por mais que fundamentais para a obtenção e metrificação de uma análise quantitativa sobre a usabilidade das ferramentas, não são suficientes por si só para realizar uma observação completa do valor que elas geram aos usuários. Por este motivo, foi realizada também uma breve entrevista semi-estruturada com cada participante após utilizarem os sistemas em questão para obter uma análise mais qualitativa, questionando sobre os seguintes pontos:

- Maiores dificuldades durante o uso.
- Satisfação com o resultado final.
- Interesse no uso da ferramenta para projetos futuros.
- Capacidade de executar por conta própria em um uso subsequente.
- Visão sobre o impacto desta tecnologia no contexto das artes visuais.

Os testes de uso em questão foram realizados com três artistas plásticos, todos com formação técnica em artes visuais pelo Instituto Federal de Pernambuco e atuantes na área a pelo menos dois anos, porém sem experiência prévia significativa no campo de programação. O número de usuários entrevistados foi escolhido por ser o mínimo sugerido por Jakob Nielsen [35] levando em consideração o custo-benefício para obter resultados pertinentes e representativos.

7.2 Resultados

Em relação aos dados coletados pelas escalas psicométricas, foram obtidos resultados mistos. A escala SUS apresentou uma média de 50,83 pontos, estando marginalmente abaixo da média para sistemas de boa usabilidade. Já no UEQ, com uma análise mais segmentada das métricas, é possível identificar os aspectos positivos e negativos da experiência de usuário com maior clareza, estando estes presentes na Figura 7.1.

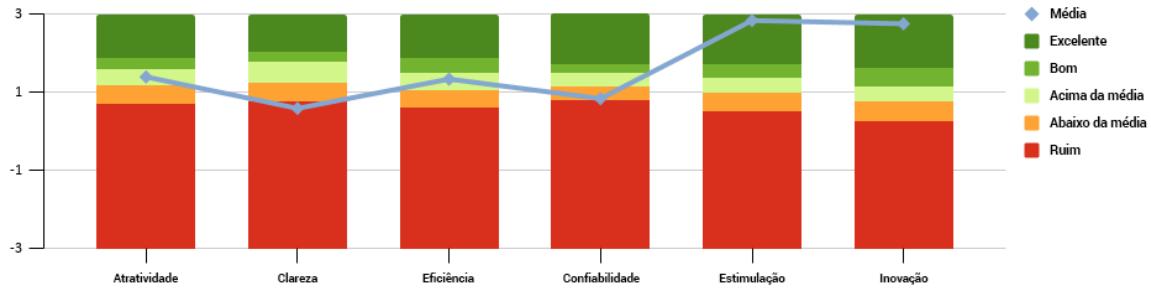


Figura 7.1: Resultados médios obtidos pelo User Experience Questionnaire. A escala vai de -3 a +3 e os intervalos são decididos por meio de estudos realizados com diversos sistemas anteriormente avaliados utilizando esta métrica.

O tempo necessário para cada participante utilizar todo o sistema, constituído pelos três *notebooks*, teve uma média de cerca de uma hora de duração. Deste, aproximadamente um terço foi usualmente utilizado pelos usuários realizando iterações na etapa de exploração, que não tem um final exato.

As entrevistas realizadas ao final do experimento tiveram duração média de aproximadamente quinze minutos, contemplando os pontos já mencionados além de qualquer comentário que o participante desejasse compartilhar por vontade própria. Juntamente com as respostas obtidas a partir da questão aberta aplicada no final do formulário, foi possível a agregação de pontos de análise qualitativos sobre o sistema.

Em geral, todos os usuários apresentaram uma boa satisfação com os resultados obtidos, alegando que a ferramenta seria útil para eles. Outro consenso foi na maior dificuldade de uso, na qual identificaram a complexidade e poluição visual da interface do Google Colaboratory como maior problema. Ainda assim, todos se consideraram aptos a utilizar o sistema por conta própria futuramente após esta primeira execução. Dois dos três participantes demonstraram interesse direto em utilizar a ferramenta para projetos futuros, mencionando alguns casos de uso no qual gostariam de aplicá-la durante o decorrer do experimento.

O resultado mais relevante dos experimentos, porém, foi o simples fato de que todos os participantes conseguiram realizar as etapas propostas em tempo hábil sem a necessidade de qualquer conhecimento prévio sobre as ferramentas. Isto, combinado com o *feedback* bastante positivo recebido em relação aos resultados alcançados durante a etapa de exploração, prova que o treinamento e exploração do espaço latente das GANs podem ser acessíveis para o público não-técnico, mesmo que a interface atualmente utilizada ainda não seja a ideal.

7.3 Discussão

Durante os experimentos com os usuários, ficou claro que o maior ponto de fricção para o uso da ferramenta esteve na interface do Colaboratory, especialmente durante as duas primeiras etapas (pré-processamento e treinamento). A quantidade de informações e recursos desnecessariamente presentes na interface por vezes foram suficientes para distrair o foco do fluxo ideal.

Mesmo com a presença dos Jupyter Widgets, as opções de customizações dentro da plataforma provaram-se limitadas, deixando poucas oportunidades para melhorias nesse quesito.

Ainda assim, todos os usuários foram capazes de realizar com sucesso as três etapas necessitando de uma quantia mínima de ajuda externa. Após esta primeira execução, porém, houve consenso de que conseguiriam utilizar outras vezes por conta própria. É possível que a apresentação prévia de um simples tutorial em vídeo exemplificando uma execução poderia ser suficiente para sanar as dúvidas que foram levantadas, eliminando assim a necessidade de ajuda externa. Tal resultado definitivamente não teria sido alcançado sem os *notebooks* devido à quantidade de programação envolvida, que seria inviável sem que tivessem esse tipo de conhecimento.

O uso do sistema de arquivos do Google Drive por meio de interfaces textuais se mostrou como uma dificuldade comum entre todos os participantes. Para o usuário não-técnico, não é normal utilizar e mover arquivos em diferentes diretórios, além de não estarem acostumados com a nomenclatura para definir caminhos. Talvez fosse mais direto pedir que criassem diretórios específicos antes da execução e usar caminhos pré-definidos para diminuir a quantidade de entradas requisitadas pelos *notebooks*.

Houve também um ponto que causou confusão na maioria dos testes, sendo este na etapa de treinamento quando é requisitada do usuário uma entrada diferente a depender se está executando pela primeira vez ou retomando um treinamento anterior. O texto estava confuso, sendo muito extenso e contendo informações desnecessárias sobre *transfer learning*. Antes do teste com o último usuário, o texto foi reescrito deixando mais clara tal diferença por meio de uma estrutura de tópicos. Com isso, o último participante foi o único que não apresentou dificuldades na etapa, provando que ainda existe espaço para melhorar as instruções e deixá-las mais acessíveis com mais iterações.

Os resultados obtidos na etapa de exploração, por sua vez, foram extremamente satisfatórios e encorajadores. Todos os usuários confirmaram uma melhora incremental em suas percepções sobre as imagens geradas durante as iterações do algoritmo genético e obtiveram populações que refletiam seus gostos e visões artísticas em poucos minutos utilizando a ferramenta. As imagens geradas pelos mesmos modelos durante o uso de cada artista mostraram-se vastamente distintas, provando que a interação humana estava ativamente trabalhando em conjunto com a máquina.

Em alguns casos, usuários citaram o tipo de estilos (cores, formas, padrões, etc) que gostariam de gerar previamente ao uso da ferramenta e conseguiram com sucesso levar a rede a sintetizar exemplares com as características buscadas. Em um dos testes, o usuário conseguiu inclusive identificar detalhes referentes a um artista específico (em uma rede treinada com obras de arte moderna) e fazer com que estes se propagassem durante as próximas iterações, misturando-se com cores nunca utilizadas pelo artista em questão durante sua carreira para gerar algo único propositalmente. Casos assim comprovam a eficácia do método escolhido para navegar por um espaço n-dimensional de maneira direcionada por um agente humano.

Realizando uma análise dos resultados provenientes das escalas psicométricas, é notável que a média calculado pelo SUS ficou distante de ser considerada ideal. Muito disso se deve aos problemas já citados relacionados ao Colaboratory, uma vez que esse questionário foca fortemente em usabilidade e não no valor ofertado ao usuário pela ferramenta.

Ademais, em relação ao UEQ, ficam evidentes as duas maiores forças do trabalho: estimulação e inovação. Estes pontos corroboram com a intuição que havia sido formada durante os testes, nos quais era perceptível que os usuários estavam constantemente surpreendidos pelos resultados alcançados e motivados para continuar seguindo em frente. Também foi possível identificar as categorias de atratividade e eficiência estando acima da média, algo bastante positivo dadas as limitações existentes nesses quesitos.

Duas das métricas, porém, ficaram abaixo da média nesta avaliação. A primeira delas, clareza, tem muita relação com os problemas de usabilidade já identificados causados pela interface utilizada e pela escrita de alguns dos textos. Já a segunda, confiabilidade, apareceu de forma inesperada pois os usuários não pareciam duvidar da solução ou da metodologia durante os experimentos. Ao explorar um pouco mais as respostas, porém, foi possível perceber que algumas das perguntas não refletiram muito bem o conceito do projeto nesta categoria. Muitas respostas classificaram o sistema como “imprevisível” de maneira positiva, relacionada às surpresas encontradas na etapa de exploração, mas acabaram contribuindo negativamente para a pontuação do UEQ uma vez que imprevisibilidade geralmente não é usualmente desejada em sistemas.

Por fim, é possível extrair conclusões a partir das respostas deixadas pelos usuários no campo aberto que havia ao final do formulário. De maneira geral, pareceu haver um forte consenso de que a ferramenta agrupa imenso valor ao auxiliar na resolução de alguns dos maiores problemas para artistas, como os bloqueios criativos e falta de inspirações. Outro ponto muito levantado foi que a interface atual é mais assustadora do que difícil, sugerindo que fosse movida para um ambiente mais amigável. Além disso, foram levantadas algumas ideias dos próprios participantes sobre como esta ferramenta poderia ser aplicada em suas obras futuras, indicando que o uso durante o experimento de fato serviu como inspiração.

CAPÍTULO 8

Conclusão

Neste trabalho foi proposto e desenvolvido um novo agente de apoio à inspiração para artistas utilizando como base a rede StyleGAN2-ADA. Além disso, foram propostas maneiras para possibilitar o treinamento deste tipo de rede de maneira acessível para usuários que não possuam conhecimento prévio de programação e hardware específico.

A interpretação dos resultados obtidos sugere um potencial significativo da exploração do espaço latente como uma promissora ferramenta para ser utilizada no meio artístico, podendo ser aplicada de diversas maneiras para auxiliar na criação de obras e coleções construídas a partir da interação entre humano e máquina. Este é um passo importante para democratizar o uso de GANs neste contexto, assim impulsionando cada vez mais o seu potencial devido às diversas visões artísticas que as poderão utilizar.

Também houve resultados encorajadores no objetivo de trazer acessibilidade ao processo de treinamento da GAN para o público não-técnico, uma vez que todos os usuários que participaram dos experimentos foram capazes de treinar com sucesso suas redes e sentiram-se confiantes de conseguir realizar o procedimento novamente no futuro graças à estrutura fornecida. Esta, por sua vez, demonstrou consideráveis problemas de usabilidade devido principalmente à interface do Google Colaboratory, mas já foi um grande passo considerando que, sem ela, tais redes seriam completamente inacessíveis para o público em questão.

8.1 Trabalhos Futuros

Para um desenvolvimento posterior deste projeto, seria necessário sanar primeiramente os problemas de usabilidade existentes por meio da criação de uma interface web própria, planejada em conjunto com uma equipe de design de UX/UI para garantir a melhor experiência possível para os usuários. Nesta arquitetura, seria necessário também um servidor para realizar o processamento em nuvem e uma infraestrutura que o disponibilizasse com placas de vídeo da Nvidia devido aos requisitos da StyleGAN.

A viabilidade deste futuro sistema em um contexto de larga escala teria que ser estudada mais a fundo, uma vez que esta poderia tornar-se financeiramente custosa dado que a etapa de treinamento das redes demanda o uso contínuo das GPUs disponíveis ao servidor. Por outro lado, caso a implementação optasse por focar apenas na etapa de exploração do espaço latente, esta não seria uma limitação significativa pois a síntese de imagens ocorre rapidamente e não exige muito processamento na GPU.

Além disso, seriam altamente agregadoras novas tentativas de implementação das técnicas discutidas neste trabalho em outras redes, sendo a CAN [24] uma excelente candidata em

potencial devido ao seu viés inherentemente artístico.

Por fim, seria de grande relevância um estudo do uso desta ferramenta pela ótica de pesquisadores da área das artes visuais, possivelmente trazendo uma visão mais ampla dos possíveis impactos desta nova tecnologia e de possíveis melhorias que poderiam ser aplicadas para aprimorar o projeto.

Referências Bibliográficas

- [1] Aican. <https://www.aican.io/>. Acessed: 2021-04-17.
- [2] Flickr. <https://www.flickr.com/services/developer/api/>. Acessed: 2021-04-21.
- [3] Freabuckler. <https://www.freabuckler.com/>. Acessed: 2021-04-10.
- [4] Freagan. <https://twitter.com/dvsch/status/1255885874560225284>. Acessed: 2021-04-10.
- [5] Ganvas studio. <https://ganvas.studio/>. Acessed: 2021-04-17.
- [6] Google colaboratory. <https://colab.research.google.com/>. Accessed: 2021-04-03.
- [7] Google drive. <https://drive.google.com/>. Accessed: 2021-04-03.
- [8] Google meet. <https://meet.google.com/>. Acessed: 2021-04-18.
- [9] Helena sarin artworks. <https://aiartists.org/helena-sarin>. Acessed: 2021-04-17.
- [10] Jupyter notebook. <https://jupyter.org/>. Accessed: 2021-04-03.
- [11] Jupyter widgets. <https://ipywidgets.readthedocs.io/>. Acessed: 2021-04-11.
- [12] Krrrl drawings gan. <http://krrrl.blogspot.com/2020/08/runway-ml-3rd-model-based-on-long-poses.html>. Acessed: 2021-04-11.
- [13] Making anime faces with stylegan. <https://www.gwern.net/Faces#stylegan-2>. Acessed: 2021-04-11.
- [14] Modern art gan. <https://twitter.com/MichaelFriesel0/status/1213697331427545088>. Acessed: 2021-04-11.
- [15] Stylegan2-ada pytorch colab fork. <https://github.com/woctezuma/stylegan2-ada-pytorch/tree/google-colab>. Acessed: 2021-04-07.
- [16] This person does not exist. <https://thispersondoesnotexist.com/>. Acessed: 2021-03-28.

- [17] Tkinter. <https://docs.python.org/3/library/tkinter.html>. Acessed: 2021-04-10.
- [18] Memo Akten. Learning to see. <http://www.memo.tv/works/learning-to-see/>. Acessed: 2021-04-17.
- [19] David Alvarez-Melis and Judith Amores. The emotional gan : Priming adversarial generation of art with emotion. 2017.
- [20] Gary Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.
- [21] John Brooke. Sus: A quick and dirty usability scale. *Usability Eval. Ind.*, 189, 11 1995.
- [22] Hans Brouwer. Audio-reactive latent interpolations with stylegan. <https://wavefun.xyz/audio-reactive-stylegan>. Acessed: 2021-04-17.
- [23] Sofia Crespo. Neural zoo. <https://neuralzoo.com/>. Acessed: 2021-04-17.
- [24] Ahmed Elgammal, Bingchen Liu, Mohamed Elhoseiny, and Marian Mazzone. Can: Creative adversarial networks, generating "art"by learning about styles and deviating from style norms, 2017.
- [25] Ian Goodfellow. Generative adversarial networks (gans) nips 2016 tutorial. Conference on Neural Information Processing Systems, Barcelona, 2016.
- [26] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.
- [27] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. *Journal of Machine Learning Research - Proceedings Track*, 9:297–304, 01 2010.
- [28] Ernst Haeckel. *Kunstformen der Natur*. Verlag des Bibliographischen Instituts, 1899. <https://www.biodiversitylibrary.org/bibliography/102214>.
- [29] HATRA (ハトラ) . Syn feather sweater / strelitzia. <https://hatroid.net/collections/synthetic-feather/products/syn-feather-sweater-strelitzia>. Acessed: 2021-04-17.
- [30] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data, 2020.
- [31] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks, 2019.

- [32] Jamshed Khan. Stylegan: Use machine learning to generate and customize realistic images. <https://heartbeat.fritz.ai/stylegans-use-machine-learning-to-generate-and-customize-realistic-images-c943388dc672>. Acessed: 2021-04-18.
- [33] Bettina Laugwitz, Theo Held, and Martin Schrepp. Construction and evaluation of a user experience questionnaire. volume 5298, pages 63–76, 11 2008.
- [34] Yisroel Mirsky and Wenke Lee. The creation and detection of deepfakes. *ACM Computing Surveys*, 54(1):1–41, Mar 2021.
- [35] Jakob Nielsen. Guerrilla hci: using discount usability engineering to penetrate the intimidation barrier. 1994.
- [36] Justin Pinkney. Toonify youself. <https://www.justinpinkney.com/toonify-yourself/>. Acessed: 2021-04-18.
- [37] Asir Saeed, Suzana Ilić, and Eva Zangerle. Creative gans for generating poems, lyrics, and metaphors, 2019.
- [38] Othman Sbai, Mohamed Elhoseiny, Antoine Bordes, Yann LeCun, and Camille Couprie. Design: Design inspiration from generative networks, 2018.
- [39] Daniel Shiffman. *The Nature of Code*. Daniel Shiffman, 2012.
- [40] Karl Sims. Galápagos. <http://www.karlsims.com/galapagos/>. Acessed: 2021-04-15.
- [41] Karl Sims. Artificial evolution for computer graphics. In *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '91, page 319–328, New York, NY, USA, 1991. Association for Computing Machinery.
- [42] Anne Spalter. Unknown departure. <https://superrare.co/artwork-v2/unknown-departure-16212>. Acessed: 2021-04-17.
- [43] Lei Wand, Wei Chen, Wenjia Yang, Fangming Bi, and Fei Yu. A state-of-the-art review on image synthesis with generative adversarial networks. *IEEE Access*, PP:1–1, 03 2020.
- [44] Yuguang Zhang. (non-)human. <https://www.ygzhang.com/non-human.html>. Acessed: 2021-04-18.