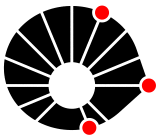
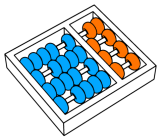




# Comparativo entre Métodos de Avaliação para Modelos de Geração de Imagens por Redes Neurais Adversárias Cíclicas

*Felipe Escórcio de Sousa* *Hélio Pedrini*

Relatório Técnico - IC-PFG-23-12
Projeto Final de Graduação
2023 - Junho

UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE COMPUTAÇÃO

# Comparativo entre Métodos de Avaliação para Modelos de Geração de Imagens por Redes Neurais Adversárias Cíclicas

Felipe Escórcio de Sousa* Hélio Pedrini†

**Resumo**

Este trabalho busca reproduzir e comparar algumas técnicas de transferência de estilo, textura e características de imagens entre classes distintas. O estudo procurou explorar algumas técnicas e métodos de transferência de estilo baseados em conhecidos modelos de aprendizado de máquina não supervisionado para tentar observar formas de se gerar imagens que reproduzam estilos artísticos e texturas, bem como mimetizem classes ou que automatizem a geração de imagens esquemáticas. Para tal tarefa, um modelo baseado em CycleGAN foi implementado utilizando camadas de ResNet18 e bases de dados de alguns pintores e estilos artísticos famosos.

## 1 Introdução

Recentemente, diversas técnicas de geração de imagens utilizando aprendizado de máquina, visão computacional e inteligência artificial têm surgido, com diversas motivações e problemas específicos a serem abordados e explorados, tornando-se um tema bastante ativo.

Um dos problemas mais tradicionais que têm sido abordados é o de transferência de texturas, estilo e geração de imagens que mimetizem características específicas de algum objeto a partir de imagens não necessariamente relacionadas com o objeto a ser imitado. Este é um problema de relativa complexidade, a depender do objetivo desejado e da qualidade a ser alcançada.

Geração de imagens unindo características de duas ou mais classes de imagens não é uma tarefa trivial de se trabalhar utilizando métodos tradicionais. Mesmo métodos mais modernos de aprendizado de máquina e inteligência artificial muitas vezes também não são simples de executar, sendo muitas vezes custosas em ambos os casos.

O sistema visual humano é bastante adaptado e torna-se difícil ocultar certos detalhes (artefatos). O mesmo não é necessariamente verdadeiro para um sistema de inteligência artificial, a depender do seu grau de desenvolvimento no cumprimento da tarefa a ela designada e, mesmo com razoável desenvolvimento, o olho humano ainda tem capacidades superiores de discernimento e observação de detalhes e padrões mais complexos e subjetivos que as máquinas não possuem.

Uma estratégia é unir uma técnica de inteligência artificial que gera imagens e uma que tenta discernir imagens falsas de imagens autênticas (que não foram geradas ou modificadas) de forma adversária, fazendo-as evoluir juntamente em suas tarefas, objetivando vencer sua adversária a ponto de gerar modelos que sejam igualmente bons de geração de imagens (geradores) e também de distinção de imagens reais (discriminadores) a ponto de ser possível enganar um observador humano, mesmo que pouco atento. Essa ideia se provou bastante eficiente em gerar modelos capazes de enganar mesmo a humanos ou imitar de forma satisfatória a produção de imagens que até então eram produzidas apenas por humanos ou algoritmos de mapeamento muito específicos. Esta forma de

*Instituto de Computação, Universidade Estadual de Campinas, Campinas, SP, 13083-852.
†Instituto de Computação, Universidade Estadual de Campinas, Campinas, SP, 13083-852.

se treinar e gerar modelos geradores e discriminadores são conhecidas por Redes Neurais Artificiais Generativas (*Generative Artificial Networks* - GANs) [5].

Por ter uma eficiência que métodos baseados na geometria ou simples aplicação de filtros não têm, torna-se possível, por exemplo, aplicar transformações em apenas algumas partes de uma imagem, sem antes necessariamente realizar uma tarefa de segmentação, tendo efeito mais local e ainda gerar imagens com formas mais sutis e classes intermediárias mais visualmente semelhantes às reais.

A capacidade das GANs de gerar imagens próximas às da realidade chega a se tornar, inclusive, de grande preocupação, tendo sido usadas para se criar as chamadas *deep fakes* com objetivos de difamação e calúnia contra pessoas e instituições públicas e privadas. Uma GAN também pode ser usada para se obter um modelo razoável que determina se as imagens são reais ou geradas, desde que treinadas com bases de dados semelhantes às usadas para criar o gerador. Apesar disso, pode-se melhorar os geradores, já que estes se aprimoram conforme os discriminadores também melhoram.

Este trabalho visa explorar uma forma de produzir uma GAN cíclica, denominada cycleGAN [22], que realiza uma operação circular de transformação, gerando uma falsificação da classe desejada e então produzindo uma versão falsificada dessa tentando retornar à classe original da qual ela partiu. Portanto, essa implementação conta com um par de geradores e um par de discriminadores diferentes para que haja um ciclo.

Por simplicidade, a implementação dos geradores utiliza uma arquitetura baseada nas camadas intermediárias da rede ResNet18 [6], utilizando 9 dessas camadas para processar imagens de tamanho 256×256 pixels como sugerido no artigo original. Também seria possível utilizar de forma satisfatória 6 camadas para processamento de imagens de 128×128 pixels.

As bases de dados de imagens foram obtidas publicamente por meio do repositório da Universidade de Berkeley, o qual é mencionado no artigo original da CycleGAN [22].

## 2 Trabalhos Relacionados

Há alguns anos, redes neurais convolucionais (*Convolutional Neural Networks* - CNNs) têm sido uma das principais ferramentas nos campos de aprendizado de máquina profundo e visão computacional, aplicadas principalmente a tarefas de segmentação, classificação, reconhecimento de objetos, entre outras. Para o problema de reconhecimento e classificação de objetos em imagens, a implementação da principal rede utilizada neste projeto é a ResNet18 [6], que procurou prover uma solução eficiente e viável para uma competição de visão computacional promovida anualmente pela ImageNet e se provou mais eficaz do que uma das redes mais populares da época, a rede VGG [16]. Seu principal diferencial em relação às outras redes é a aplicação de camadas residuais intermediárias que provêm a ela maior facilidade em otimização e menor complexidade, apesar de possuir mais camadas do que as outras redes eficientes da época.

A solução de Gatys et al. [4] utiliza algumas camadas da VGG19 [16] para realizar a extração de estilo e conteúdo para mesclagem. A VGG [16] é uma rede interessante para essa tarefa, pois consiste de várias camadas de filtros pequenos seguidos por camadas de ativação, o que permite a ela imitar filtros maiores, também diminuindo a dimensionalidade dos vetores no processo.

Ainda neste tema, há também estudos que procuram explorar o problema a partir do uso de AutoEncoders Variacionais (em inglês, *Variational AutoEncoders* - VAEs) [11]. Embora não seja um tópico diretamente abordado neste trabalho, mas em resumo, é uma solução que gera novos dados compactando características e as projetando em um espaço vetorial de menor dimensão e depois o projetando na dimensão desejada, tentando manter o máximo de informação original. Normalmente, esta solução pode ser bem mais rápida e adequada para uso em produção do que outras.

A solução que buscamos abordar aqui emprega redes adversárias, mais especificamente, tentando explorar uma implementação que utiliza o princípio de ciclicidade de tradução da transformação entre

classes, ou seja, a tradução de uma classe para outra deve ser a mesma se traduzida de volta para a classe original.

Há outras abordagens que utilizam a mesma ideia básica das GANs, tais como Pixel2Pixel [8], CoGAN [2], PixelGAN [12], StyleGAN [9], BigGAN [1] e SAGAN [21] (em que a BigGAN [1] é baseada). que também já foram utilizadas para a mesma tarefa ou outras tarefas de geração de imagens com objetivos semelhantes de combinação de características subjetivas e geração de novas imagens.

A rede Pixel2Pixel[8] é a principal motivadora para a criação das CycleGANs [22] e é uma rede que realiza uma tradução aproximadamente direta de um pixel ao outro, utilizada, por exemplo, em tarefas de transformação de imagens de satélite em imagens de mapas e de desenhos para imagens de objetos reais, transformação de preto e branco para colorido e vice-versas. É uma rede adversária condicional (*Conditional GAN* - cGAN) [13], que resumidamente é uma rede que, durante seu treinamento, recebe os rótulos das entradas para que ela aprenda a gerar imagens de classes determinadas e não somente imagens genéricas aleatórias possibilitando então ter mais controle sobre a saída desejada pela rede. O artigo que propõe esse modelo para tradução de imagens também introduz o uso de uma rede discriminadora chamada PatchGAN [8].

Dos mesmos autores da CycleGAN, tem-se também um trabalho mais recente que utiliza recortes (*patches*) para manter consistência na tradução das imagens (*Contrastive Unpaired Translation* - CUT [14]), buscando manter informações consistentes entre partes das imagens tratadas e que conseguiu obter resultados melhores, entretanto, cujo método não será abordado neste projeto.

# 3 Materiais e Métodos

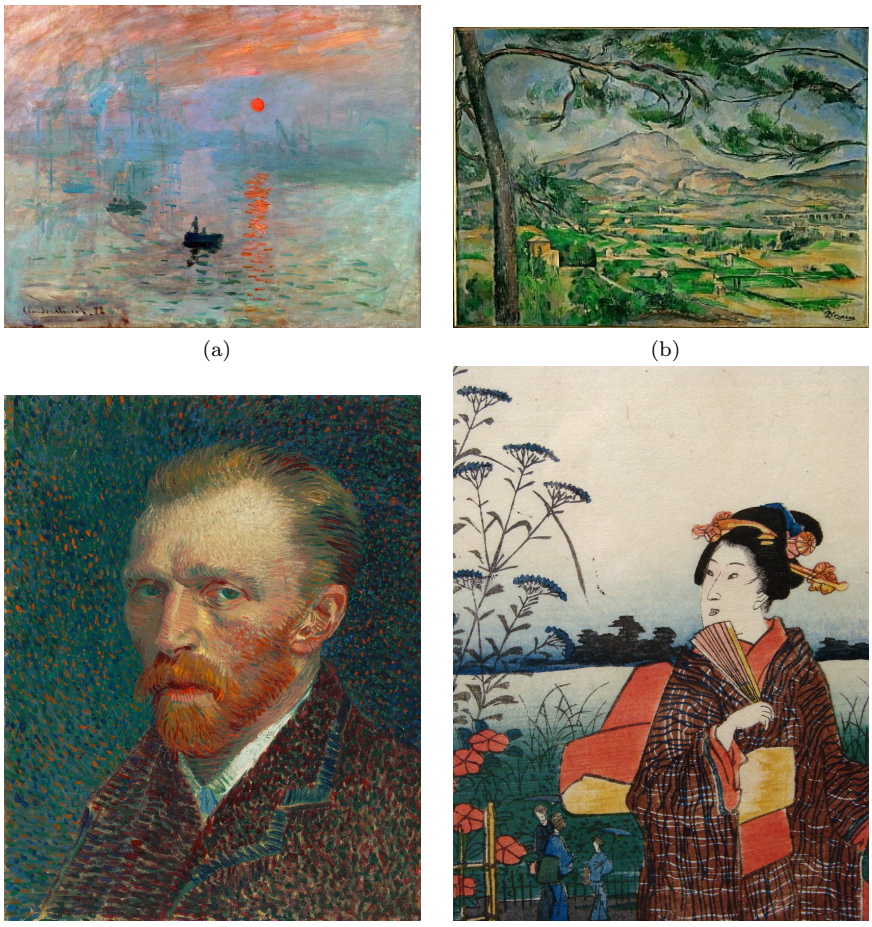
Os conjuntos de dados utilizados foram bases de obras de pintores tais como Monet (pintor impressionista francês), Cezanne (pintor pós-impressionista francês), Van Gogh (pintor pós-impressionista holandês), diversas gravuras do estilo Ukyio-ê (estilo popular de pintura por xilogravura no Japão entre os séculos XVII e XIX) e fotografias de paisagens disponibilizadas no repositório da Universidade de Berkeley [22].

Todos os conjuntos consistem de diversas imagens com dimensões tamanho 256×256 pixels e coloridas. Todas as imagens foram diretamente utilizadas, sem a aplicação de uma pré-processamento. A Figura 1 ilustra exemplos de imagens contidas nos conjuntos de dados.

As redes foram treinadas no *Google Collab* utilizando a plataforma com *upgrade* para o Pro+ para permitir treinamento suficiente e mais rápido do que seria possível em relação à plataforma utilizada de forma gratuita. Nessa versão, geralmente são disponibilizadas instâncias de placas gráficas (GPUs) NVIDIA V100 ou NVIDIA A100 e 90 GBytes de memória.

A rede foi implementada utilizando-se da biblioteca PyTorch para Python 3.6, por ser uma biblioteca bastante conhecida e com compatibilidade e eficiência no uso de GPUs, além de prover uma interface razoavelmente simples para se trabalhar com esse recurso.

Por ser um problema de resultado subjetivo, a tarefa de quantificar o resultado é bastante difícil, sendo originalmente realizada por pesquisa com público e por uma breve análise visual dos resultados. Neste projeto, procurou-se realizar a extração de métricas utilizando um algoritmo chamado Inception Score [15], que procura aferir uma nota do quão realista é a produção do modelo, se ele é capaz de gerar imagens diferentes e se há semelhanças com classes de objetos reais. É um método que se comporta razoavelmente semelhante ao julgamento do olho humano, sendo muito utilizado para avaliar os diferentes resultados das GANs. Sua implementação é baseada em um modelo de redes neurais profundas para tarefa de reconhecimento desenvolvido pela Google chamado InceptionV3 [18], inicialmente como um módulo para a GoogLeNet [17] e desenvolvido para o desafio de reconhecimento de imagens do ImageNet. O modelo pré-treinado é gerado utilizando-se por volta de 30 mil imagens e pode ser treinado para um número arbitrário de imagens e de classes de imagens.



(a) (b) (c) (d)

Figura 1: Exemplos de obras de (a) Monet, (b) Paul Cezanne, (c) Vincent Van Gogh e (d) uma xilogravura do estilo Ukiyo-ê.

O resultado é a distribuição de probabilidade sobre as classes da ImageNet. O valor mínimo a ser obtido é 1,0 (uma única classe) e o número de classes utilizadas no treinamento do modelo (aqui 1000 classes).

O funcionamento dessa métrica leva em conta a entropia e a entropia condicional encontrada nas imagens, avaliando a diversidade e a qualidade das imagens obtidas. A entropia mede a diversidade de imagens geradas, que quando elevada indica menor quantidade de informações. A entropia condicional é obtida a partir da probabilidade marginal, calculando-se a entropia de cada imagem dada a partir da distribuição de todas as imagens geradas. Entropias marginais de valores menores indicam imagens geradas mais fáceis de classificação do que valores mais elevados. O valor final é a média geométrica entre os valores das entropias.

$$\text{IS} = \exp(\mathbb{E}_{x \sim p_{data}(x)}[D_{KL}(p(y|x)||p(y))]) \tag{1}$$

em que $x$ é uma imagem ou um vetor multidimensional, $p_{data}(x)$ é a distribuição real das imagens, $p(y|x)$ a distribuição das classes de objeto dada uma imagem $x$ e $p(y)$ é a margem de distribuição das classes de objeto

Entretanto, o Inception Score [15] também apresenta alguns problemas advindos do fato de ser uma rede geralmente treinada sobre a base ILSVRC 2014. Alguns desses problemas são que o classificador está limitado à quantidade de classes presentes no conjunto de treinamento, o *score* é bastante relacionado às texturas e não às formas devido à natureza das CNNs e que, para tarefas de geração de imagens de classes específicas, geralmente o *score* é pouco representativo. Esses problemas não serão críticos para a nossa tarefa. Um possível problema é que o *score* possa ser associado à variedade de imagens presentes no conjunto de imagens utilizadas para teste.

Outra métrica utilizada como auxílio ao IS [15] é o FID (*Fréchet Inception Distance*) [7], que considera a qualidade das imagens além da diversidade como métrica. A ideia por trás da métrica FID é utilizar o cálculo de distâncias entre os conjuntos de imagens (geradas e reais) a partir da média e a matriz de covariância entre os grupos extraída a partir de camadas da InceptionV3 [18] (outra rede neural convolucional criada pela Google para classificação de imagens e treinada sobre a ImageNet). Como medida de distância, valores menores indicam melhor qualidade, ou seja, maior proximidade.

A métrica FID é calculada como:

$$\text{FID} = ||\mu_1 - \mu_2||_2^2 + Tr(\Sigma_1 + \Sigma_2 - 2(\Sigma_1\Sigma_2)^{1/2}) \tag{2}$$

em que $\mu_1$ e $\mu_2$ são os vetores de média das distribuições de características da imagem gerada e da imagem real, respectivamente. Os parâmetros $\Sigma_1$ e $\Sigma_2$ são as matrizes de covariância das distribuições de características da imagem gerada e da imagem real, $Tr$ é a função traço, que retorna a soma dos elementos na diagonal principal de uma matriz e $||\cdot||_2$ é a norma L2, que retorna a raiz quadrada da soma dos quadrados dos elementos do vetor.

É importante notar que, apesar dessas métricas, a avaliação humana ainda é de grande e indispensável importância. Não utilizaremos essas métricas aqui por serem mais subjetivas e requererem tempo e uma população razoável para realizar o estudo estatístico adequado.

## 3.1 Arquitetura

A implementação de redes neurais adversárias requer o estabelecimento de dois tipos de redes que farão parte: um gerador e um discriminador, que atuarão em um jogo semelhante ao de polícia e ladrão, em que um deles tenta criar falsificações dos artefatos (o gerador) e o outro tenta determinar se o objeto passado é ou não uma falsificação. Esse jogo pode ser modelado por uma estratégia min-max, tal que:

$$\min_G \max_D V(G,D) = \mathbb{E}_{x\sim p_{data}(x)}[logD(x)] + \mathbb{E}_{z\sim p_z(x)}[log(1 - D(G(z))] \tag{3}$$

em que $p_g$ é a distribuição do gerador sobre os dados denotados por $x$ e $p_z$ o mapeamento dos dados de $G(z;\theta_g)$, sendo $G$ uma função diferenciável que modela o perceptron multicamada gerador de $\theta_g$ parâmetros iniciais. A rede de perceptrons multicamada discriminadora pode ser modelado por uma função denominada $D(z;\theta_d)$ de parâmetros iniciais $\theta_d$ que representa a probabilidade de $x$ ser autêntico ou pertencer a ($p_g$. O papel de $G$ é tentar maximizar sua capacidade de gerar imagens que pareçam autênticas, $D$ tenta, por outro lado, minimizar esse resultado.

Como o objetivo da CycleGAN [22] é realizar uma transformação $X \to Y$ e também mapear do domínio $Y$ de volta para o domínio $X$, tal que evite haver divergências entre as duas traduções, de maneira que uma operação seja a inversa da outra. Denotam-se $G : X \to Y$ e $F : Y \to X$ como as operações. Para cada uma também é definido um discriminador próprio $D_Y$ e $D_X$, discriminadores no domínio do resultado. Tem-se então, a equação da função de perda:

$$\min_G \max_{D_Y} V(G, D_Y, X, Y) = \mathbb{E}_{y\sim p_{data}(y)}[\log(D_Y(y))] + \mathbb{E}_{x\sim p_{data}(x)}[\log(1 - D_Y(G(x))] \tag{4}$$

em que se pode denotar por $(\mathcal{L}_{cyc}(G, D_Y, X, Y)$, e similarmente:

$$\min_F \max_{D_X} V(F, D_X, Y, X) = \mathbb{E}_{x \sim p_{data}(x)}[\log(D_X(x))] + \mathbb{E}_{y \sim p_{data}(y)}[\log(1 - D_X(F(y))] \quad (5)$$

também denotado por $(\mathcal{L}_{cyc}(F, D_X, Y, X)$.

É possível fazer com que mapeamentos $G$ e $F$ produzam resultados mapeados distribuídos por $X$ e $Y$ em uma possível permutação de imagens no domínio desejado que satisfaçam a distribuição desejada, entretanto, perdas adversárias sozinhas não são suficientes para garantir que uma entrada individual $x_i$ seja mapeado para uma saída desejada $y_i$. É então induzida a necessidade da utilização de uma forma de tornar essa propriedade mais consistente para o objetivo de mapeamento e, para isso, é introduzido o conceito de ciclicidade (Figura 2), tal que $x \to G(x) \to F(G(x)) \approx x$, assim como também satisfazer a consistência cíclica inversa $y \to F(y) \to G(F(y)) \approx y$.

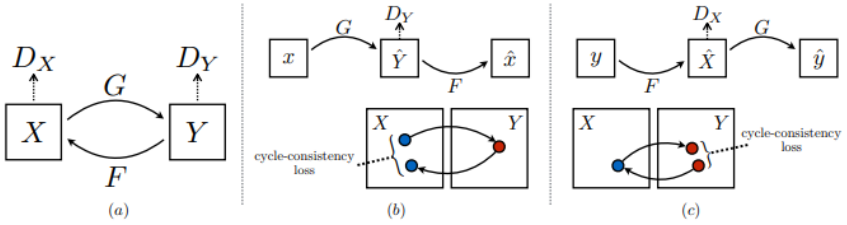



Figura 2: Mapeamentos de $G$ para $F$ e de $F$ para $G$, ciclicamente [22].

Esse comportamento é induzido a partir da função de perda de consistência cíclica (*cycle consistency loss*):

$$\mathcal{L}_{cyc}(G, F) = \mathbb{E}_{x \sim p_{data}(x)}[\|F(G(x)) - x\|_1] + \mathbb{E}_{y \sim p_{data}(y)}[\|G(F(x)) - y\|_1] \quad (6)$$

O gerador (Figura 3) foi construído sequencialmente com, primeiramente, a criação uma borda, uma convolução para aumento das dimensões, normalização e retificação (ReLU) para preparação da imagem, duas camadas de *downsampling* (novamente com uma convolução, uma normalização e uma retificador), 9 camadas residuais semelhante às camadas intermediárias da ResNET19, 2 camadas de upsampling (semelhantes às de *downsampling*, mas com objetivo contrário) e, por fim, uma camada com uma borda, uma convolução e a ativação final (escolhida a função de tangente hiperbólica). Utilizamos 9 camadas residuais para imagens de tamanho 256×256 pixels, tendo sido possível também em estados iniciais de implementação experimentar uma versão com apenas 6 camadas em imagens de 128×128 pixels.

A rede discriminadora (Figura 4) é um modelo baseado em uma PatchGAN [8], uma rede neural que determina a autenticidade de uma imagem a partir de trechos independentemente entre si para um mapa de $N \times N$ de vetores de saída $X$ a partir de uma média das ativações dos trechos. Em resumo, ela é uma rede que aplica convoluções sobre a imagem, uma rede convolucional comum que não retorna apenas um escalar, sendo eficiente em termos de memória e processamento. Uma grande vantagem dessa arquitetura é a de, por avaliar imagens mais localizadamente, levar o gerador a gerar imagens de melhor qualidade.

A partir dessas duas redes, busca-se minimizar a quantidade de erro entre as gerações e as predições de um conjunto e outro, simultaneamente, conforme o ciclo. Utilizou-se como função principal de erro a *MeanSquareError* (erro de quadrados mínimos) por se tratar de um problema

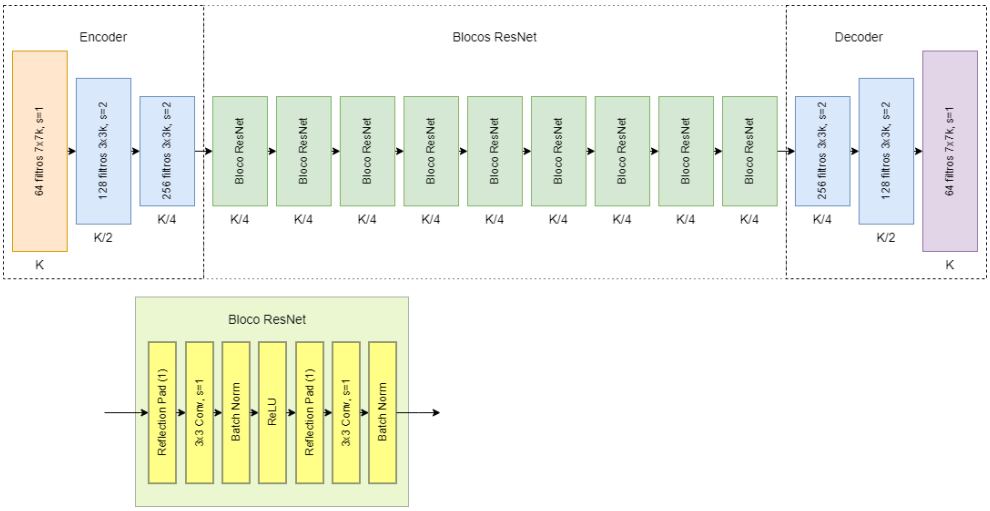



Figura 3: Arquitetura do gerador baseado em blocos da ResNet19. Fonte: Autor.

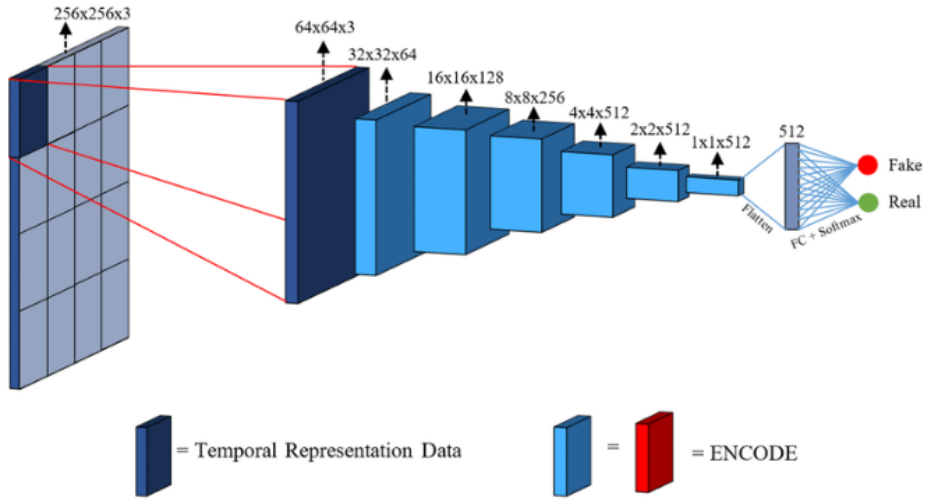



Figura 4: Arquitetura da *PatchGAN* [19].

de regressão e não necessariamente de classificação de imagens, além de discriminá-las entre verdadeira ou falsa. A diminuição ou tendência de redução das perdas é um critério para se observar a convergência dos modelos criados neste trabalho.

# 4 Resultados

A Figura 5 ilustra alguns resultados obtidos em imagens de paisagens, tema muito comum aos artistas que geraram as imagens do conjunto de dados e extraídas da Internet e licenciadas por *Creative Commons*.

Pode-se observar que, para paisagens, o resultado é satisfatório e que, além de preservar as formas


Figura 5: Imagens de paisagens. À esquerda, as originais e, nas próximas colunas, as mesmas imagens quando aplicados os modelos treinados com obras de Monet, Cezanne, Van Gogh e xilogravuras Ukiyo-ê, respectivamente.

e alterar as cores, ele também adiciona alguns efeitos que tornam as imagens mais semelhantes a uma pintura, como o efeito de pinceladas aplicadas às telas. No caso da rede treinada para obras Ukiyo-ê, pode-se notar que objetos como nuvens se tornam bem menos distinguíveis em relação ao céu.

A Figura 6 ilustra alguns resultados dos modelos testados para imagens de objetos. Os resultados são semelhantes e pode-se destacar melhor como alguns contrastes são realçados ou amenizados. Algumas formas são um pouco mais borradas e as cores transformadas e fica evidente como essas cores servem como um filtro de cada artista/estilo devido a sua intensa presença em suas obras e nos conjuntos de dados utilizados.

A Figura 6 ilustra alguns resultados obtidos em imagens contendo pessoas. Os resultados deixam bastante a desejar nesse tipo de imagens, não apresentando grandes mudanças em relação às


Figura 6: Imagens de objetos (maçã [10], caneca [3] e vaso [20]). À esquerda, as originais e, nas próximas colunas, as mesmas imagens quando aplicados os modelos treinados com obras de Monet, Cezanne, Van Gogh e xilogravuras Ukiyo-ê, respectivamente.

imagens originais, aparentando apenas a aplicação de filtros simples de cor, temperatura, saturação e iluminação. Pode-se notar tambpem que, para esses artistas/estilos, as figuras humanas são muitas vezes representadas de maneira bastante diversa e estilizada, sendo que para ter resultados mais interessantes provavelmente seria necessário o uso de outros conjuntos de dados (o que não seria possível no caso dos artistas famosos).

A Figura 8 ilustra os resultados obtidos em uma fotografia. Pode-ser observar que os modelos acabaram perdendo definição de imagem, tornando os resultados mais borrados, menos detalhados e de formas mais difusas, menos iluminados e também por acabar tendo paletas de cores mais limitadas em relação ao modelo treinado por Zhu et al. [22], demonstrando assim, um sobreajuste (*overfiting*) em relação a este.

A Figura 9 ilustra a evolução das funções de perda e da métrica FID das redes geradoras durante o processo de treinamento e como elas se alteram durante os ciclos de treinamento e o que podem significar.

Observa-se que, das funções de perda utilizadas no treinamento das redes, apenas a *cycle loss* apresenta uma convergência visível, em declínio, que é de fato o parâmetro que se deseja minimizar. As outras funções de perda apresentam uma certa convergência, que apesar de haver picos, é menos pronunciada e perceptível em relação à *cycle loss* que está se buscando minimizar. Um outro gráfico interessante é o de FID que, por um lado do ciclo, tem picos inconstantes, enquanto, por outro, tem picos bem menores e permanece quase constantemente e relativamente baixo em relação ao outro. Os gráficos de outros modelos seguem as mesmas tendências.

A Tabela 1 compara alguns resultados obtidos neste trabalho com resultados obtidos em outros trabalhos correlatos que investigam CycleGANs [22].


Figura 7: Imagens contendo pessoas. À esquerda, as originais e, nas próximas colunas, as mesmas imagens quando aplicados os modelos treinados com obras de Monet, Cezanne, Van Gogh e xilogravuras Ukiyo-ê, respectivamente.

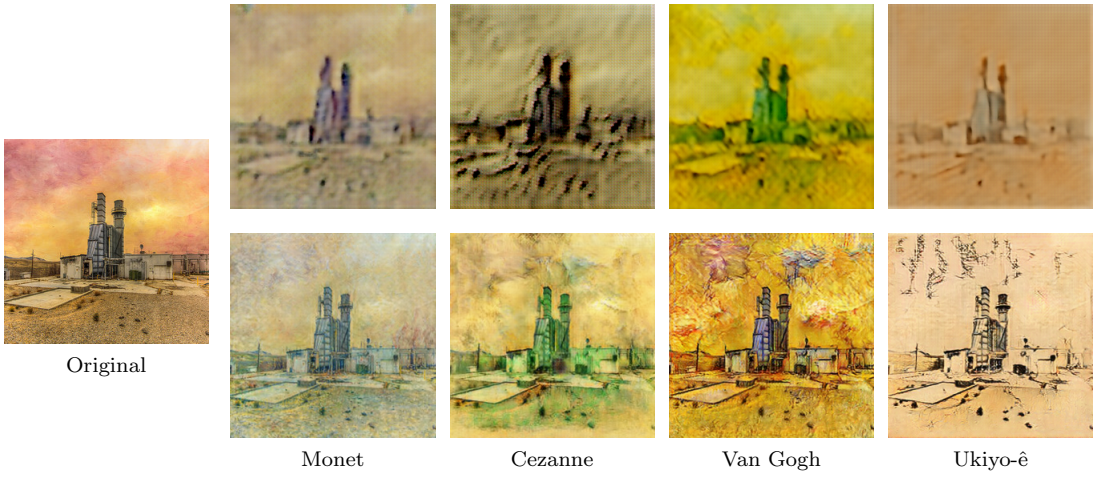
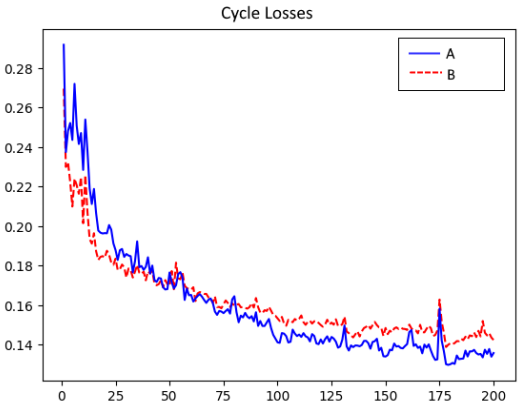


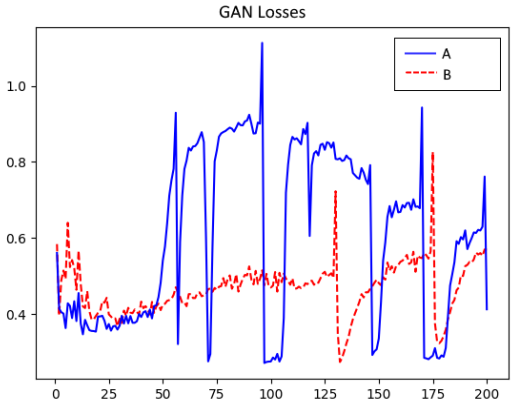

Figura 8: Comparação de uma fotografia com as transformações realizadas pelo modelo do experimento (anterior) e pelo modelo pré-treinado de Zhu et al. [22].

Observa-se valores razoavelmente próximos de Inception Score, mas não exatamente iguais, possivelmente devido à perda de qualidade das imagens nas transformações das redes do experimento.

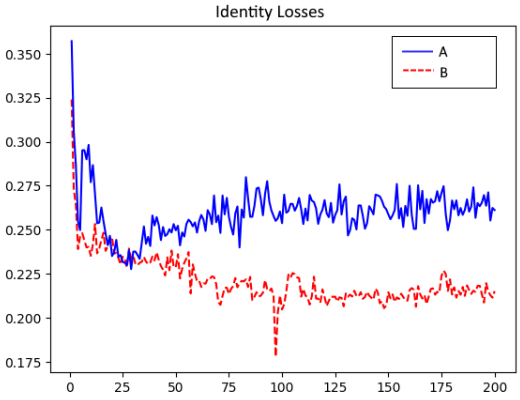
É interessante observar que algumas métricas convergem de forma razoavelmente semelhante conforme há a progressão do treinamento das redes, enquanto outras não convergem, o que é apontado pelos autores do projeto original [23]. A métrica que era esperada convergir é a *cycle loss*. Por não ser uma transformação tão intensa nas imagens, é compreensível que a métrica FID não tenha convergido e tenha às vezes oscilado em algumas épocas durante o treinamento.



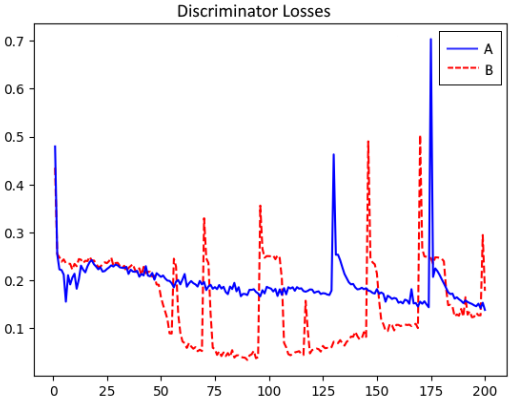

(a) Cycle Losses entre A (função $G$) e B(função $F$)



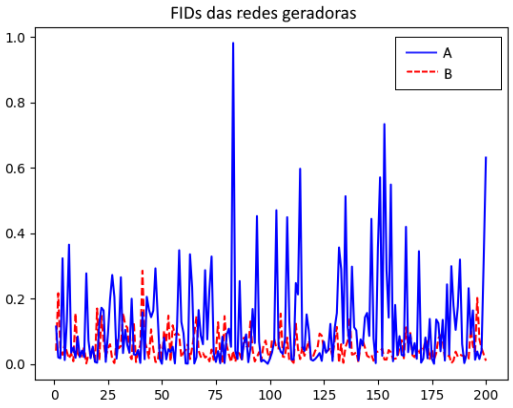

(b) GAN Losses entre A (função $G$) e B(função $F$)




(c) Identity Losses entre A (função $G$) e B(função $F$)




(d) Discriminator Losses entre A (função $G$) e B(função $F$)




(e) FIDs entre A (função $G$) e B(função $F$)

Figura 9: Gráficos comparativos entre as duas fases do ciclo, A(Função $G$) e B(função $F$)

| | Experimento | Pré-treinado |
|---|---|---|
| *Ground-truth* | 16.18 | |
| Monet | 10.83 | 11.53 |
| Cezanne | 9.21 | 10.17 |
| Van Gogh | 9.76 | 7.84 |
| Ukiyo-ê | 9.73 | 9.09 |

Tabela 1: Comparação entre os resultados aplicando-se o *Inception Score* ao conjunto de teste sem aplicar modelos e aplicando-se os modelos do experimento e do autor.

# 5 Conclusões

Neste trabalho, um modelo foi reproduzido que satisfatoriamente imita o estilo de alguns pintores, embora não atingindo um grau em que se possa realmente persuadir um observador que tenha grande conhecimento em pintura expressionista e pós-expressionista.

A tarefa de imitar grandes gênios das artes com redes simples é bastante desafiadora, sendo necessário um nível de refinamento e sofisticação muito maior, caso se almeje um resultado mais próximo de uma pintura a óleo ou xilogravura (que por ter formas normalmente menos convencionais é mais complexo de imitar). Muitas vezes, as redes aprenderam a sobrepor e destacar algumas cores mais presentes nas obras dos artistas.

A partir da comparação realizada, observou-se que ocorreu a convergência dos modelos e dos parâmetros esperados durante o treinamento e sua evolução, assim como a não convergência de outros como também era esperado, obtendo-se ainda resultados visuais razoáveis. Valores interessantes foram alcançados para as métricas FID.

Como propostas para trabalhos futuros, sugere-se a exploração de novas técnicas de aprendizado de máquina profundo, o uso de outras métricas de avaliação dos resultados e novos conjuntos de dados.

# Referências

[1] Andrew Brock, Jeff Donahue e Karen Simonyan. *Large Scale GAN Training for High Fidelity Natural Image Synthesis*. 2019. arXiv: `1809.11096 [cs.LG]`.

[2] *Coupled Generative Adversarial Networks, author=Ming-Yu Liu and Oncel Tuzel*. 2016. arXiv: `1606.07536 [cs.CV]`.

[3] Jamal Fanaian. *Ubuntu Mug*. Último acesso em junho de 2023. 2010. URL: `https://openverse.org/image/d9df2f21-2f06-46f5-b6e4-af8da83a2f4b?q=mug`.

[4] Leon A. Gatys, Alexander S. Ecker e Matthias Bethge. "Image Style Transfer Using Convolutional Neural Networks". Em: *IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 2414–2423.

[5] Ian J. Goodfellow et al. *Generative Adversarial Networks*. 2014. arXiv: `1406.2661 [stat.ML]`.

[6] Kaiming He et al. *Deep Residual Learning for Image Recognition*. 2015. arXiv: `1512.03385 [cs.CV]`.

[7] Martin Heusel et al. *GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium*. 2018. arXiv: `1706.08500 [cs.LG]`.

[8] Phillip Isola et al. *Image-to-Image Translation with Conditional Adversarial Networks*. 2018. arXiv: 1611.07004 [cs.CV].

[9] Tero Karras, Samuli Laine e Timo Aila. *A Style-Based Generator Architecture for Generative Adversarial Networks*. 2019. arXiv: 1812.04948 [cs.NE].

[10] kirinohana. *Camera Test Apple*. Último acesso em junho de 2023. 2013. URL: https://openverse.org/image/13303ef2-8ed1-4b52-8c05-94d9ba7252a7?q=apples.

[11] Zhi-Song Liu, Vicky Kalogeiton e Marie-Paule Cani. *Multiple Style Transfer via Variational AutoEncoder*. 2021. arXiv: 2110.07375 [cs.CV].

[12] Alireza Makhzani e Brendan Frey. *PixelGAN Autoencoders*. 2017. arXiv: 1706.00531 [cs.LG].

[13] Mehdi Mirza e Simon Osindero. *Conditional Generative Adversarial Nets*. 2014. arXiv: 1411.1784 [cs.LG].

[14] Taesung Park et al. *Contrastive Learning for Unpaired Image-to-Image Translation*. 2020. arXiv: 2007.15651 [cs.CV].

[15] Tim Salimans et al. *Improved Techniques for Training GANs*. 2016. arXiv: 1606.03498 [cs.LG].

[16] Karen Simonyan e Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2015. arXiv: 1409.1556 [cs.CV].

[17] Christian Szegedy et al. *Going Deeper with Convolutions*. 2014. arXiv: 1409.4842 [cs.CV].

[18] Christian Szegedy et al. *Rethinking the Inception Architecture for Computer Vision*. 2015. arXiv: 1512.00567 [cs.CV].

[19] Supavadee Aramvith Thittaporn Ganokratanaa e Nicu Sebe. *The PatchGAN Structure in the Discriminator Architecture*. Último acesso em junho de 2023. 2020. URL: https://www.researchgate.net/figure/The-PatchGAN-structure-in-the-discriminator-architecture_fig5_339832261.

[20] Marco Verch. *A Composition with Vintage Photo Frame and Vase with Dry Flowers*. Último acesso em junho de 2023. 2017. URL: https://openverse.org/image/520a1460-7d56-486a-85ef-8e63e626230c?q=flower%5C%20vase.

[21] Han Zhang et al. *Self-Attention Generative Adversarial Networks*. 2019. arXiv: 1805.08318 [stat.ML].

[22] Jun-Yan Zhu et al. *Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks*. 2020. arXiv: 1703.10593 [cs.CV].

[23] Ju-Yan Zhu. *pytorch-CycleGAN-and-pix2pix - Frequently Asked Questions*. Último acesso em junho de 2023. 2018. URL: https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix/blob/master/docs/qa.md.