Pontifícia Universidade Católica
do Rio de Janeiro

Pedro Henrique Barroso Gomes

FCGAN: Convoluções Espectrais via Transformada Rápida de Fourier para Campo Receptivos de Abrangência Global em Redes Adversárias Generativas

Dissertação de Mestrado

Dissertação apresentada como requisito parcial para a obtenção do grau de Mestre pelo Programa de Pós–graduação em Informática, do Departamento de Informática da PUC-Rio .

Orientador: Prof. Marcelo Gattass

Rio de Janeiro
Março de 2024

Pontifícia Universidade Católica
do Rio de Janeiro

Pedro Henrique Barroso Gomes

FCGAN: Convoluções Espectrais via Transformada Rápida de Fourier para Campo Receptivos de Abrangência Global em Redes Adversárias Generativas

Dissertação apresentada como requisito parcial para a obtenção do grau de Mestre pelo Programa de Pós–graduação em Informática da PUC-Rio . Aprovada pela Comissão Examinadora abaixo:

Prof. Marcelo Gattass
Orientador
Departamento de Informática – PUC-Rio

Prof. José Alberto Rodrigues Pereira Sardinha
PUC-Rio

Dr. Jan Jose Hurtado Jauregui
PUC-Rio

Dr. Ítalo de Oliveira Matias
IUCAM

Rio de Janeiro, 5 de Março de 2024

Pedro Henrique Barroso Gomes

Graduou-se em Engenharia da Computação pela Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio), onde trabalha na mentoria de projetos de inovação para a indústria no Instituto ECOA PUC-Rio.

Ficha Catalográfica

Barroso Gomes, Pedro Henrique

FCGAN: Convoluções Espectrais via Transformada Rápida de Fourier para Campo Receptivos de Abrangência Global em Redes Adversárias Generativas / Pedro Henrique Barroso Gomes; orientador: Marcelo Gattass. – 2024.

76 f: il. color. ; 30 cm

Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, 2024.

Inclui bibliografia

1. Informática – Teses. 2. Geração de Imagens. 3. Campo Receptivo Global. 4. Domínio da Frequência. 5. Redes Adversárias Generativas. I. Gattass, Marcelo. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

Aos meus pais e minha família
pelo apoio e encorajamento.

Agradecimentos

Agradeço aos meus pais, Roberto e Maristela, e a minha avó, Vilma, pelo apoio incondicional.

À minha barulhenta e calorosa família. Aos meus amigos, que tenho muita sorte de ter encontrado ao longo da vida. À Vivian, por ser meu porto seguro durante todo esse processo.

Ao professor Marcelo Gattass, meu orientador. A Luiz Fernando, meu coorientador, pelo incentivo e apoio constante. Ao professor Jônatas Wehrmann, pelos ensinamentos na etapa inicial deste trabalho.

Agradeço aos também a todos meus colegas de Instituto ECOA PUC-Rio, pelos anos de crescimento e companheirismo. Aos amigos que conheci nesses anos de PUC-Rio, que me apoiaram com debates, conselhos, e risadas.

Agradeço todas as dificuldades, contratempos, renúncias e ideias malsucedidas. E sou grato por todo o crescimento e aprendizado que essa jornada me proporcionou.
O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

Resumo

Barroso Gomes, Pedro Henrique; Gattass, Marcelo. **FCGAN: Convoluções Espectrais via Transformada Rápida de Fourier para Campo Receptivos de Abrangência Global em Redes Adversárias Generativas**. Rio de Janeiro, 2024. 76p. Dissertação de Mestrado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Esta dissertação propõe a Rede Generativa Adversarial por Convolução Rápida de Fourier (FCGAN). Essa abordagem inovadora utiliza convoluções no domínio da frequência para permitir que a rede opere com um campo receptivo de abrangência global. Devido aos seus campos receptivos pequenos, GANs baseadas em convoluções tradicionais enfrentam dificuldades para capturar padrões estruturais e geométricos. Nosso método utiliza Convoluções Rápidas de Fourier (FFCs), que usam Transformadas de Fourier para operar no domínio espectral, afetando globalmente os canais da imagem. Assim, a FCGAN é capaz de gerar imagens considerando informações de todas as localizações dos mapas de entrada. Essa nova característica da rede pode levar a um desempenho errático e instável. Mostramos que a utilização de normalização espectral e injeções de ruído estabilizam o treinamento adversarial. O uso de convoluções espectrais em redes convolucionais tem sido explorado para tarefas como inpainting e super-resolução de imagens. Este trabalho foca no seu potencial para geração de imagens. Nossos experimentos também sustentam a afirmação que features de Fourier são substitutos de baixo custo operacional para camadas de self-attention, permitindo que a rede aprenda informações globais desde camadas iniciais. Apresentamos resultados qualitativos e quantitativos para demonstrar que a FCGAN proposta obtém resultados comparáveis a abordagens estado-da-arte com profundidade e número de parâmetros semelhantes, alcançando um FID de 18,98 no CIFAR-10 e 38,71 no STL-10 - uma redução de 4,98 e 1,40, respectivamente. Além disso, em maiores dimensões de imagens, o uso de FFCs em vez de self-attention permite batch-sizes com até o dobro do tamanho, e iterações até 26% mais rápidas.

Palavras-chave

Geração de Imagens; Campo Receptivo Global; Domínio da Frequência; Redes Adversárias Generativas.

Abstract

Barroso Gomes, Pedro Henrique; Gattass, Marcelo (Advisor). **FCGAN: Spectral Convolutions via FFT for Channel-Wide Receptive Field in Generative Adversarial Networks**. Rio de Janeiro, 2024. 76p. Dissertação de Mestrado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

This thesis proposes the Fast Fourier Convolution Generative Adversarial Network (FCGAN). This novel approach employs convolutions in the frequency domain to enable the network to operate with a channel-wide receptive field. Due to small receptive fields, traditional convolution-based GANs struggle to capture structural and geometric patterns. Our method uses Fast Fourier Convolutions (FFCs), which use Fourier Transforms to operate in the spectral domain, affecting the feature input globally. Thus, FCGAN can generate images considering information from all feature locations. This new hallmark of the network can lead to erratic and unstable performance. We show that employing spectral normalization and noise injections stabilizes adversarial training. The use of spectral convolutions in convolutional networks has been explored for tasks such as image inpainting and super-resolution. This work focuses on its potential for image generation. Our experiments further support the claim that Fourier features are lightweight replacements for self-attention, allowing the network to learn global information from early layers. We present qualitative and quantitative results to demonstrate that the proposed FCGAN achieves results comparable to state-of-the-art approaches of similar depth and parameter count, reaching an FID of 18.98 on CIFAR-10 and 38.71 on STL-10 - a reduction of 4.98 and 1.40, respectively. Moreover, in larger image dimensions, using FFCs instead of self-attention allows for batch sizes up to twice as large and iterations up to 26% faster.

Keywords

Image Generation; Global Receptive Field; Frequency Domain; Generative Adversarial Networks.

Sumário

Lista de figuras

Lista de tabelas

Lista de Abreviaturas

GAN – Generative Adversarial Network

G – Generator

D – Discriminator

CNN – Convolutional Neural Network

RGB – Red, Green, Blue

BN – Batch Normalization

SN – Spectral Normalization

SNGAN – Spectrally Normalized Generative Adversarial Network

SAGAN – Self Attention Generative Adversarial Network

FCGAN – Fourier Convolution Generative Adversarial Network

FFC – Fast Fourier Convolution

FFT – Fast Fourier Transform

DFT – Discrete Fourier Transform

FID – Fréchet Inception distance

ISC – Inception Score

CGI – Computer Graphic Imagery

ViT – Vision Transformer

ERF – Effective Receptive Field

TRF – Theoretical Receptive Field

When you come out of the storm, you won't be the same person who walked in. That's what this storm's all about.

Haruki Murakami, *Kafka On The Shore.*

1
Introdução

A construção de modelos por meio de métodos de Deep Learning apresentou um novo paradigma para o campo de visão computacional. Capacitados pelo poder de múltiplas camadas de processamento, esses modelos computacionais conseguem de capturar padrões e estruturas sofisticadas presentes em largas escalas de dados (VOULODIMOS et al., 2018). O aumento do interesse pela utilização de Deep Learning se deu ao passo que diversas arquiteturas de redes neurais se mostraram capazes de superar o então estados-da-arte em uma série de tarefas relacionadas a imagens, vídeos, e modelos 3d.

No mercado e na sociedade, tarefas de visão computacional são utilizadas em um conjunto significativo de soluções de alto impacto. A fim de realizar a análise de imagens médicas, por exemplo, é estudado o uso de redes neurais profundas para a tarefa de classificação de imagens. Abordagens nesse contexto tiveram sucesso no reconhecimento de Alzheimer baseado em imagens de ressonâncias magnéticas, classificação de catarata nuclear a partir de exames com lâmpada de fenda, detecção de tuberculose em exames radiológicos do tórax (LITJENS et al., 2017).

Softwares de edição e restauração de fotos também se apresentam como meios de utilização cotidianas de modelos de visão computacional. A remoção de objetos indesejados, manchas, ou rasgos é comumente associada à tarefa de inpainting de imagens (ELHARROUSS et al., 2020). No problema de inpainting, assume-se que a imagem entregue teve parte de seu conteúdo removido. Essa remoção pode se dar pelo tempo (degradação) ou pelo utilizador do software (no caso da seleção do objeto a ser removido). A tarefa do algoritmo de inpainting é reconstruir a parte faltante do conteúdo da imagem (GUILLEMOT; MEUR, 2013).

No mercado do entretenimento, as possibilidades decorrentes da exploração da visão computacional têm se expandido consideravelmente. Modelos generativos de imagens podem ser utilizados por estúdios de filmes para o rejuvenescimento ou envelhecimento de atores. *Deep fakes* podem ser utilizados no processo de dublagem. Alterações em cenários podem ser realizadas após o processo de gravação (ANANTRASIRICHAI; BULL, 2022). Essas aplicações são exemplos da tarefa de geração de imagem, e representam um avanço significativo na manipulação visual.

Para a geração de imagens, as Redes Generativas Adversárias (GANs) (GOODFELLOW et al., 2014) têm demonstrado obter resultados notáveis em

uma variedade de arquiteturas, sendo as redes convolucionais profundas especialmente bem-sucedidas. No entanto, GANs enfrentam desafios consideráveis relacionados à estabilidade do treinamento e ao alto custo computacional de treinamento. Além disso, os campos receptivos locais das convoluções tradicionais podem se tornar uma razão para que os modelos tenham dificuldade em capturar padrões geométricos e estruturais nas imagens sintetizadas.

Para abordar essa limitação no campo de visão das redes, pesquisadores introduziram a SAGAN (ZHANG et al., 2018), que incorpora operadores de self-attention (VASWANI et al., 2017) para modelar dependências de longo alcance. Desde então, mecanismos de attention continuam sendo empregados em várias arquiteturas. Em trabalhos recentes, Vision Transformers (ViT) utilizam o mecanismo de multi-headed attention para melhorar a qualidade das imagens, especialmente para classes de imagens que possuem padrões estruturais ou texturas mais predominantes (LEE et al., 2021).

Ao mesmo tempo, para outras tarefas de visão computacional, como classificação e inpainting de imagens, novas arquiteturas aproveitam os operadores de Convoluções Rápidas de Fourier (FFC)(CHI; JIANG; MU, 2020) para obter campos receptivos que enxergam toda a imagem de uma vez em suas convoluções. O FFC realiza convoluções no domínio da frequência com o uso da Transformada Rápida de Fourier. Essa abordagem inovadora capacita as operações com uma percepção global dos canais da imagem, levando essas redes a superar o estado da arte em seus respectivos campos, como classificação e inpainting de imagens, e detecção de objetos e poses em imagens e vídeos.

Isso levanta um questionamento natural. Qual é o efeito da aplicação de convoluções no domínio da frequência nas GANs para a tarefa de geração de imagens? O presente trabalho visa estudar a aplicação convoluções espectrais como uma alternativa de menor custo computacional para capturar padrões de longo alcance para a geração de imagens. É proposta a Rede Generativa Adversarial por Convoluções de Fourier (FCGAN, na sigla em inglês), uma arquitetura que substitui convoluções convencionais por operadores de Convolução Rápida de Fourier.

A dissertação deu origem a um artigo aceito na IFIP International Conference on Artificial Intelligence Applications & Innovations (AIAI).

Este documento está estruturada em seis capítulos. No Capítulo 1, foi feita a introdução do contexto do trabalho, sua motivação e seus objetivos. No Capítulo 2, são apresentados trabalhos relacionados que tem como objetivo (i) a utilização de redes neurais profundas para a tarefa de geração de imagens e (ii) a utilização de features no domínio da frequência em visão computacional. A fundamentação teórica utilizada para a construção dessa dissertação está

contida no Capítulo 3. O Capítulo 4 detalha a metodologia empregada e discorre sobre o método proposto e suas principais características. Detalhamos os experimentos realizados no Capítulo 5, para, por fim, discorrer sobre as conclusões dos experimentos e potenciais trabalhos futuros no Capítulo 6.

2
Trabalhos relacionados

Esse capítulo está dividido em três seções. A primeira aborda a utilização de redes neurais para a tarefa de geração de imagens. Na segunda seção, são explorados usos do domínio da frequência em tarefas de visão computacional. Esse uso se caracteriza como uma potência desde as abordagens mais tradicionais do campo até modelos profundos recentes. A terceira seção apresenta o módulo de Fast Fourier Convolutions (CHI; JIANG; MU, 2020). É apresentado um conjunto de redes neurais que utilizaram esses operadores para atingir resultados estado-da-arte em suas respectivas tarefas.

2.1
Uso de Redes Neurais Profundas para geração de imagens

Uma das principais abordagens para a sintetização de novas imagens por meio de modelos de inteligência artificial é o uso de Redes Adversárias Generativas (GANs) (GOODFELLOW et al., 2014), em especial aliados às características de Redes Neurais Convolucionais (CNNs) propostas por LeCun et al. (LECUN et al., 1998). Desde de sua concepção, essa arquitetura foi amplamente explorada para incluir elementos que favorecessem uma melhor sintetização de imagens. Esse formato de redes neurais profundas dominou o estado da arte até recentemente, tendo seu desempenho equiparado pelos novos modelos de Denoising Diffusion (HO; JAIN; ABBEEL, 2020).

A DCGAN (RADFORD; METZ; CHINTALA, 2016) se tornou um marco como uma das primeiras redes que incorporou com sucesso a arquitetura de uma rede convolucional profunda dentro do framework de redes generativas adversárias. Sua arquitetura trouxe alterações como a eliminação de camadas totalmente conectadas, a troca de camadas de pooling por convoluções com stride, e o uso de Batch Normalization. Entretanto, um de seus principais problemas era a instabilidade no treinamento, devido à natureza adversarial do treinamento de seu gerador e discriminador. A SNGAN (MIYATO et al., 2018) buscou trazer estabilidade para esse modelo com mudanças na arquitetura que introduziram a Spectral Normalization no discriminador.

Ainda que alcançassem bons resultados em conjuntos de dados como dígitos e rostos alinhados, redes com arquiteturas mais tradicionais encontraram dificuldade na construção de imagens de classes com uma grande natureza geométrica ou texturas predominantes (ZHANG et al., 2018). Devido à maneira que essas redes realizam os cálculos para extração de features, padrões globais

da imagem só passam a ser entendidos em camadas mais profundas. Isso faz com que essas informações sejam perdidos ao sintetizar novas estruturas.

Uma das principais abordagens para o reconhecimento de dependências em longas distâncias é a utilização do mecanismo de self-attention. Na rede SAGAN (ZHANG et al., 2018), os mecanismos de attention foram introduzidos em camadas estratégicas do gerador e do discriminador. Essa inclusão entregou resultados superiores ao então estado-da-arte, mostrando que essa proposta foi efetiva em reconhecer mais padrões no dados observados. Os módulos de self-attention possuem um maior equilíbrio na troca entre o custo computacional e a eficiência estatística em relação a outras abordagens propostas. Entretanto, trabalhos recentes têm apontado o alto peso computacional da self-attention, sendo substituída com sucesso pela utilização de Transformadas de Fourier para tarefas de processamento de linguagem natural (LEE-THORP et al., 2021) e classificação de imagens (RAO et al., 2021).

A partir de arquiteturas como a ProGAN (KARRAS et al., 2018), as redes passaram a ser capazes de gerar imagens fotorealistas. A ProGAN traz como principal ideia uma nova metodologia de treinamento com o crescimento progressivo do gerador e discriminador. A família das StyleGANs (KARRAS; LAINE; AILA, 2019) aplicou melhorias para o gerador da ProGAN, trazendo o atual estado-da-arte para geração de imagens com redes adversárias generativas.

A sua versão mais recente, a StyleGAN3 (KARRAS et al., 2021), buscou resolver o problema de que o processo de síntese em GANs típicas dependem demais das coordenadas absolutas dos pixels. Isso faz com que apareçam artefatos - elementos que geram estranheza - quando ocorram rotações ou translações em imagens geradas, o que expõe as imagens como falsas. Para isso, são utilizadas features de Fourier ao invés da entrada constante aprendida da StyleGAN2 (KARRAS et al., 2020). São amostradas frequências de forma uniforme dentro de uma faixa de frequência circular, correspondendo à resolução original de entrada, e as mantemos fixas ao longo do treinamento.

A intuição por trás do uso de features de Fourier para a rotação e translação das imagens geradas é que as camadas da rede geradora têm capacidade limitada de introduzir transformações globais na imagem, assim, o valor de entrada dessa rede possui um papel cricial em definir a orientação da imagem gerada. Isso demonstra que o uso do domínio da frequência pode potencializar a geração de imagens até para modelos estado-da-arte.

2.2 Deep Learning e Domínio da Frequência

O uso do domínio da frequência e de Transformadas de Fourier é explorado comumente no contexto de imagens. Em 1971, o domínio da frequência foi utilizado para extração de features em imagens de radiografia (HALL et al., 1971). Em 1978, o módulo da transformada de Fourier de uma imagem foi utilizada para realizar a reconstrução de objetos presentes nela (FIENUP, 1978). Em 1988, foi proposto o método de Iterative Blind Deconvolution (AYERS; DAINTY, 1988) utilizando-se de Fourier para processamento de imagens.

Mas a contribuição de Transformadas de Fourier não se encerrou com algoritmos de processamento de imagem clássicos. Seu uso aliado a modelos de Deep Learning é um campo explorado desde o surgimento das primeiras arquiteturas profundas. Aproveitando-se das propriedades da transformada de Fourier, LeCun argumentou pelo uso de FFTs para o treinamento rápido de redes convolucionais (MATHIEU; HENAFF; LECUN, 2013). O treinamento é acelerado ao calcular convoluções como um produto pontual no domínio de Fourier. Com o objetivo de realizar tarefas de classificação de imagens, foi proposta uma rede convolucional treinada inteiramente no domínio da frequência: a FCNN (PRATT et al., 2017). Entre as vantagens apontadas pela rede estão (i) o aumento exponencial da eficiência em imagens de dimensões maiores e (ii) uma redução significativa de tempo de treinamento da rede.

Mais recentemente, a exploração do domínio de frequência continua a despertar o interesse de uma série de estudos em visão computacional. A distribuição espectral de imagens geradas por modelos tem sido explorada para detectar imagens falsas (CHANDRASEGARAN; TRAN; CHEUNG, 2021a) e *deep fakes* (FRANK et al., 2020). A aprendizagem de features no domínio de frequência também foi aplicada com o objeto de se atingir uma maior robustez adversarial (LI et al., 2021). Para o rendering de modelos 3d, os modelos NeRF (MILDENHALL et al., 2021) utilizam o positional embedding da entrada a partir do domínio da frequência. É argumentado que apesar de redes neurais serem aproximadores universais de funções, operar no domínio espacial apresenta resultados inferiores.

Muitos desses trabalhos se baseiam nas evidências que features de Fouries permitem que redes aprendam sinais de frequência mais altas em baixas dimensões (TANCIK et al., 2020). A proposta de positional embedding também já foi também empregada no framework de GANs. A INR-GAN (SKOROKHODOV; IGNATYEV; ELHOSEINY, 2021) se aproveita dessa premissa representar a geração de imagens na forma de *implicit neural representations* (INRs) - tratam o gerador da GAN como um Multi Layer Perceptron (MLP)

que busca prever o pixel RGB de uma determinada coordenada (x, y). Com essa abordagem, o modelo foi capaz de superar os resultados da StyleGAN2.

2.3 Fast Fourier Convolutions e suas aplicações

Observando a potência de uma compreensão geral das imagens durante o treinamento de modelos profundos, o módulo de Fast Fourier Convolution (CHI; JIANG; MU, 2020) (FFC) foi proposto como um operador que permite convoluções de campo receptivo globais. Com o objetivo de substituir convoluções regulares que ocorrem somente o domínio espacial (dos pixels), operadores FFC permitem que informações globais sejam propagadas durante toda a profundidade das redes a partir da propagação de dois ramos. *O ramo local* x^l é formado por convoluções convencionais e tem como objetivo aprender a partir da vizinhança de pixels. Já o *ramo global* x^g apresenta operações no domínio da frequência, tendo como objetivo capturar informações de longo alcance. A inclusão desses operadores em modelos de arquitetura ResNet trouxe maior eficiência para tarefas de classificação de imagens, de ações em vídeos e de pontos-chaves humanos.

Desde de sua publicação em 2020, um conjunto de arquiteturas tem adotado com sucesso os operadores de FFC. O modelo LaMa (Resolution-robust Large Mask Inpainting) (SUVOROV et al., 2021) substituiu as convoluções regulares por FFCs com o objetivo de aprimorar resultados de inpainting de imagens quando as partes retiradas das imagens eram significativas. Nos experimentos, também foi observado que os operadores FFC possuem características que tornam a rede significativamente mais robusta para resoluções mais altas que os modelos comparados. Também é apontado que a rede é capaz de alcançar essa qualidade com um número significativamente menor de parâmetros em relação a redes que utilizam o mecanismo de self-attention. Como conclusão, os autores trazem o potencial da Transformada de Fourier como uma alternativa eficiente dos módulos de self-attention ou Vision Transformers.

Buscando obter melhores resultados para a tarefa de super-resolução, o modelo NL-FFC (SINHA; MOORTHI; DHAR, 2022) utiliza blocos de FFC aliados a módulos de non-local attention para adquirir conhecimento global sobre a imagem. É apontado que o módulo de non-local attention auxilia os blocos de FFC e funciona complementarmente, em especial ao mapear features globais para contextos locais. Por fim, LoHiSC (LONG et al., 2023) aproveita o insight que as informações de cor da imagem estão predominantemente contidas na parte de baixa frequência da imagem, enquanto os detalhes de contorno e textura estão principalmente concentrados na parte de alta

frequência. O modelo utiliza uma alteração do módulo de FFC para incluir filtros de baixa e alta para o cálculo dessas features.

Em resumo, Fast Fourier Convolutions têm sido aplicadas com sucesso para alcançar resultados de ponta em tarefas de classificação, detecção de poses, inpainting de imagens com máscara extensas, super-resolução, e colorização. Como é discutido no trabalho da SAGAN (ZHANG et al., 2018), as convoluções regulares têm dificuldade em capturar dependências de longo alcance e padrões geométricos e estruturais em classes de imagens. Levando isso em consideração, nosso trabalho levanta o questionamento de se o campo receptivo global das Fast Fourier Convolutions pode beneficiar a síntese de imagens em um framework de GAN.

3
Fundamentação Teórica

Nesse capítulo são introduzidos conceitos relacionados à tarefa de geração de imagens. Em seguida, fundamentos básicos de redes neurais convolucionais são definidos. Por fim, são apresentados detalhes de arquiteturas e operadores de redes neurais profundas para geração de imagem.

3.1
Tarefa de geração de imagens

Na literatura, a tarefa de geração de imagens pode ser definida pelo processo de gerar artificialmente imagens que contenham conteúdos específicos desejados (KUMAR et al., 2020). A partir de um conjunto de dados recebido, o objetivo de uma rede é aprender os padrões e estruturas. Ela deve então utilizar esse conhecimento para produzir novas imagens visualmente plausíveis que se assemelhem à imagens reais dos dados de treinamento.

Formalmente, muitos trabalhos definem a tarefa de uma rede geradora $\mathcal{G}$ como mapear uma distribuição de probabilidade p_g que se aproxime da distribuição real dos dados $p_g = p_{dados}$ (GOODFELLOW et al., 2014; KINGMA; WELLING, 2013; BENGIO et al., 2014; ARJOVSKY; CHINTALA; BOTTOU, 2017).

Pode-se dividir as subtarefas de geração de imagem a partir das diferentes abordagens para o controlar a sintetização das características desejadas. Entre as principais abordagens estão: a geração não-condicionada, geração condicionada por classe e *text-to-image*.

3.1.1
Não-condicionado

Considere $\mathcal{X} = \mathbb{R}^{d \times d}$ o espaço das imagens originais, e uma variável aleatória Z com uma distribuição de probabilidade fixa $p(z)$. A tarefa de geração não-condicionada se refere a passar essa variável aleatória por uma função paramétrica $g_\theta : \mathcal{Z} \rightarrow \mathcal{X}$ (normalmente uma rede neural) que gera uma imagem seguindo uma certa distribuição $\mathcal{P}_\theta$ (ARJOVSKY; CHINTALA; BOTTOU, 2017). Ao variar θ, pode-se alterar a distribuição e torná-la mais próxima da distribuição real dos dados $\mathcal{P}_r$. Em muitas arquiteturas, a entrada é entregue como um ruído aleatório z amostrado de uma distribuição normal $\mathcal{N}(\mu, \sigma)$ definida com média μ e desvio padrão σ.

No treinamento não-condicionado, a função de perda da rede $\mathcal{L}(\theta)$ também não recebe informações adicionais em relação à imagem. Em algumas arquiteturas, porém, a função de perda considera a informação de se a imagem é real ou gerada. Assim, treinamento não-condicionado não um sinônimo direto de treinamento não-supervisionado. As redes apresentadas nesse trabalho, em especial as GANs, podem apresentar características simultâneas de treinamento supervisionado e não-supervisionado (GOODFELLOW et al., 2020).

Na geração não-condicionada, conjuntos de dados com grande diversidade de elementos e características são um desafio mais complexo - principalmente em redes com menor ordem de magnitude de parâmetros. Isso porque a rede não recebe informações em relação à quantidade de classes ou o valor real da classe de cada amostra, obrigando a rede a compreender essa diversidade afim de não gerar imagens híbridas entre classes.

Apesar da geração não-condicionada possuir uma implementação mais direta, ela não deixou de ser relevante com o surgimento de novos modelos. Arquiteturas em todos os níveis de complexidade, como DCGANs, SAGANs, ProGANs e StyleGANs, utilizam-se de treinamento não-condicionado. Um bom treinamento não-condicionado pode atestar a capacidade da rede geradora de representar diferentes classes em seu espaço latente.

3.1.2 Condicionado por classe

Na geração condicionada por classe, os modelos recebem um vetor de condições ou rótulos c_i junto a cada respectiva imagem x_i. Assim, os rótulos são entregues tanto no momento da síntese da imagem, por exemplo junto ao vetor de ruído z, quanto para a função de perda. Temos portanto, uma geração condicionada $G(z,\, c)$ (MIRZA; OSINDERO, 2014). O objetivo da rede é não só gerar imagens que se assemelham ao conjunto de dados real, mas também respeitar as restrições impostas pelo rótulo da classe.

As implementações mais comuns não incorporam os rótulos de classes diretamente nas funções de perda $J(\theta)$ (MIRZA; OSINDERO, 2014). Existem, porém, trabalhos que se aproveitam de arquiteturas mais complexas, como classificadores auxiliares (ODENA; OLAH; SHLENS, 2017), encarregados de identificar a classe correta das imagens geradas ao minimizar a perda $\mathcal{L}_C$ de forma:

$$\mathcal{L}_C = \mathbb{E}[\log_P(C = c\,|X_{real})] + \mathbb{E}[\log_P(C = c\,|X_{fake})]$$

A inclusão da informação da classe é muitas vezes realizada a partir de uma camada de Embedding que recebe vetores *one-hot* que representam

a classe da imagem. O resultado da camada de Embedding pode ser (i) adicionado em novas dimensões do vetor de entrada z, na forma $z \in \mathbb{R}^{Z+C}$, em que Z é a dimensão do ruído z e C é a dimensão de saída da camada de Embedding; (ii) concatenado aos canais do tensor da imagem ou do tensor da saída de alguma camada intermediária. O condicionamento por classe pode também ser realizado em camadas intermediárias, como as de normalização (VRIES et al., 2017). Inclusões de rótulos como descritas nessa subseção são de rápida implementação e adaptáveis a uma série de arquiteturas.

Figura 3.1: Exemplo de treinamento condicionado por classe. Fonte: (ZHANG et al., 2018)

3.1.3 Text-to-image

Em um treinamento condiconado por texto, temos uma rede geradora do seguinte formato $G : \mathbb{R}^Z \times \mathbb{R}^T \rightarrow \mathbb{R}^D$, em que T é a dimensão do texto e D é a dimensão da imagem (REED et al., 2016). Junto ao vetor de ruído z, é codificado o texto t utilizando um codificador de texto ϕ. Temos, portanto, a geração de uma imagem x dada por $x \leftarrow G(z, \phi(t))$.

A maneira mais direta de treinar uma rede geradora condicional é considerar pares (texto, imagem) como observações conjuntas e avaliar esses pares como reais ou falsos. Isso, porém, não observa se as imagens geradas no treinamento correspondem ao contexto da incorporação de texto. Para isso, a função de perda deve punir as imagens realistas que não condizem com o texto. Uma abordagem é incluir no treinamento exemplos de imagens reais com textos não alinhados com seu conteúdo - que devem ser tratados como imagens falsas (REED et al., 2016).

Abordagens mais recentes também utilizam modelos auxiliares. CLIP (RADFORD et al., 2021) recebe uma imagem e tem o objetivo a tarefa de

image-to-text, isto é, a geração de um texto descritivo da imagem. Funções de perda podem utilizar a métrica CLIP Score (HESSEL et al., 2021), que calcula a similaridade da descrição entregue pelo modelo CLIP e a entrada de texto dada à rede geradora.

$$\text{CLIPScore}(i, t) = w \times \max(\cos(E_i, E_t), 0) \tag{3-1}$$

O que corresponde à similaridade de cosseno do embedding visual E_i de uma imagem i entregue ao modelo CLIP, e E_t o embedding textual do CLIP de uma legenda t.

A ControlGAN (LI et al., 2019) estudou o uso de uma rede auxiliar a partir da VGG (SIMONYAN; ZISSERMAN, 2014). A VGG tem como objetivo extrair features semânticas da imagem, e a função de perda é definida como

$$\mathcal{L}_{per}(I', I) = \frac{1}{C_i H_i W_i} ||\phi_i(I') - \phi_i(I)||_2^2$$

Onde $\phi_i(I)$ é o valor da ativação da i-ésima camada da rede VGG, e H_i e W_i são a altura e largura do feature map, respectivamente.

Dada sua versalidade, muitas soluções de software utilizam a geração de imagens condicionadas por texto. É o caso de produtos como DALL-E 2 [1] e Midjourney [2]. Modelos estado-da-arte como Latent Diffusion Model (ROMBACH et al., 2022) e DALL-E (RAMESH et al., 2021) são implementados considerando essa abordagem, e trabalhos anteriores tiveram sucesso em condicionar arquiteturas como a StyleGAN (SAUER et al., 2023).

Figura 3.2: Exemplo de imagens geradas via *text-to-image*. Fonte: (QIAO et al., 2019)

3.2 CNNs e suas propriedades

Redes Neurais Convolucionais (CNNs) são uma arquitetura análoga às redes neurais tradicionais, no sentido que são formadas por um conjunto de neurônios que recebem uma entrada e realizam uma operação para entregar uma saída. A maior diferença para as redes tradicionais é o uso de operadores de convolução. Redes convolucionais frequentemente são formadas um conjunto

[1] https://openai.com/dall-e-2
[2] https://www.imagine.art

de camadas convolucionais, camadas de *poolings* e camadas densas (totalmente conectadas).

Convoluções são operadores lineares entre duas funções. Pode-se descrever a formulação matemática de uma convolução de uma imagem I e um kernel K para gerar uma imagem filtrada O pela equação abaixo:

$$O(i,j) = (K * I)(i,j) = \sum_m \sum_n K(m,n) \cdot I(i-m, j-n) \tag{3-2}$$

No contexto de redes convolucionais, o saída resultante de uma convolução a partir de um filtro recebe o nome de *feature map*. Os neurônios que performam convoluções em imagens são comumente organizados em três dimensões, as dimensões espaciais (altura e largura) e a profundidade (canais). Nas camadas ocultas da rede, os neurônios passam a receber o conjunto de *feature maps* das camadas anteriores. O aprendizado de uma camada é dado a partir de kernels, ou filtros, treináveis. Esses kernels geralmente possuem dimensões espaciais pequenas (muitas vezes 3×3 ou 4×4), mas percorrem a imagem inteira vertical e horizontalmente. As redes convolucionais, portanto, encorajam a aprendizagem de estruturas locais. Dado uma entrada $x \in \mathbb{R}^{h \times w \times d}$ uma vizinhança local $\mathcal{N}_k$ em volta do pixel x_{ij} é extraída com extensão espacial do tamanho k do kernel, resultando em uma região de tamanho $k \times k \times d$.

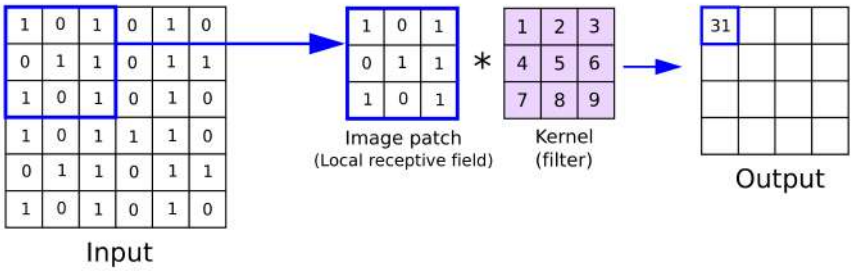

Figura 3.3: Camada de convolução em uma CNN.

As camadas de convolução podem se utilizar dos parâmetros de *padding* e *stride* para controlar a mudança de dimensionalidade do tensor de saída. O padding controla a adição de pixels (geralmente zeros) em torno das bordas de entrada antes da passagem do kernel. Já o stride controla o tamanho do passo da movimentação do kernel durante a convolução. Em redes de classificação, é comum que o modelo recebe uma imagem e entrega como saída um valor escalar. Ocorre, portanto, uma redução de dimensionalidade, ou *downscale*, da entrada ao longo da rede feita ou por camadas de *pooling* ou por convoluções com *stride*. Já redes geradoras comumente recebem um ruído (dimensão espacial 1×1) e deve retornar as dimensões de largura e

altura da imagem. Uma abordagem para realizar esse *upsample* é a convolução transposta, ou deconvolução, que utilizam os kernels treináveis para expandir o feature map recebido.

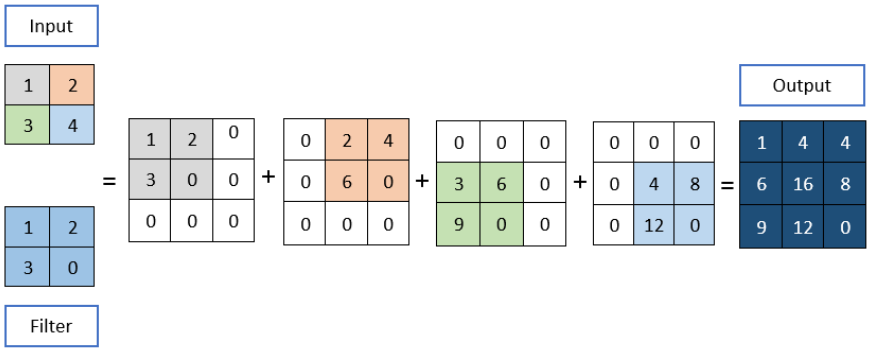

Figura 3.4: Operação de convolução transposta em uma CNN.

Devido aos kernels pequenos, uma camada convolucional recebe pouca informação global sobre estruturas e padrões presentes nas imagens, tendo ao seu alcance somente o campo receptivo local. Ainda que o empilhamento de convoluções combine os campos receptivos de camadas anteriores e entregue informações globais em camadas mais profundas da rede, trabalhos posteriores argumentaram que esse fato pode ser um dificultador para tarefas relacionadas à visão computacional (ZHANG et al., 2018; SUVOROV et al., 2021).

3.3 GANs e suas propriedades

Redes Generativas Adversárias (GANs, do inglês Generative Adversarial Networks) foram propostas em 2014 como um novo framework de modelos generativos por meio de treinamento adversário (GOODFELLOW et al., 2014). Em uma GAN, são treinados simultaneamente dois modelos: um modelo gerador, como os discutidos na Seção 3.1, e um modelo discriminador, que tem como objetivo aprender a discernir imagens reais de imagens geradas.

3.3.1 Treinamento Adversarial

Formalmente, mantém-se o objetivo do modelo gerador G de aprender a distribuição p_g que mapeia o vetor de ruído de entrada z para o espaço dos dados na forma $x = G(z)$. Define-se também um modelo discriminador $D : x \rightarrow \hat{k}$ que recebe uma imagem x e entrega um valor escalar $\hat{k}$ que representa a probabilidade da imagem x vir dos dados reais e não de p_g.

Otimizamos os pesos do discriminador θ_d minimizando a *binary cross entropy* das previsões tanto para as imagens reais quanto para as imagens geradas

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} [\log D(x_i) + \log(1 - D(\hat{x}_i))]$$

Em que m é o número de instâncias no *mini-batch*, e x_i e $\hat{x}_i$ são a i-ésima imagem selecionada da distribuição de dados reais e dos dados geradores, respectivamente.

O que caracteriza o treinamento adversarial de uma GAN é que as imagens geradas entregues à função de perda do discriminador são sintetizadas pelo gerador. Assim, o bom rendimento de um modelo implica em uma maior perda do outro, levando-os a ser treinados em simultâneo.

$$D(\hat{x}_i) = D(G(z_i))$$

Em que z_i é o ruído amostrado de Z para atual iteração. A atualização dos pesos do gerador θ_g acontece de maneira semelhante, buscando recompensá-lo quando o discriminador erra a avaliação sobre as imagens sintetizadas.

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log(1 - D(G(z_i)))$$

O treinamento da GAN é realizado, portanto, a partir de um jogo min max como definido abaixo:

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

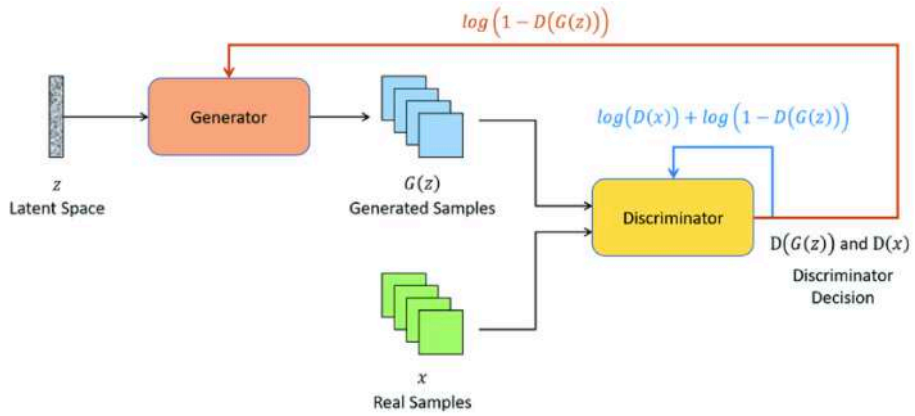

Figura 3.5: Arquitetura básica de uma GAN.

3.3.2 Arquiteturas para o gerador e discriminador

Em sua proposta inicial, tanto o gerador quanto o discriminador eram Multilayer Perceptrons formados por camadas totalmente conectadas e Dropouts. Desde de então, foram experimentadas diferentes arquiteturas para cada um desses modelos. A DCGAN (RADFORD; METZ; CHINTALA, 2016) propôs o uso de Redes Neurais Convolucionais (CNNs) tanto no discriminador quanto no gerador. Em sua arquitetura, são propostos um gerador e discriminador que empilham blocos de convolução, normalização e ativação, respectivamente duplicando ou reduzindo pela metade a dimensionalidade a cada camada. A ProGAN (KARRAS et al., 2018) trouxe a abordagem de redes treinadas progressivamente. Em vez de gerador e discriminador buscarem aprender a transformação $G : \mathbb{R}^Z \rightarrow \mathbb{R}^{128\times128}$ e $D : \mathbb{R}^{128\times128} \rightarrow \hat{k}$, para imagens 128×128 por exemplo, a ProGAN aprende primeiro em uma resolução mais baixa, por exemplo 4×4. Conforme o treinamento avança, são adicionadas camadas em G e D de resoluções maiores. Todas as camadas continuam treináveis até a última camada de alta resolução ser adicionada. Tanto na ProGAN quanto na DCGAN, as arquiteturas do gerador e discriminador são espelhadas.

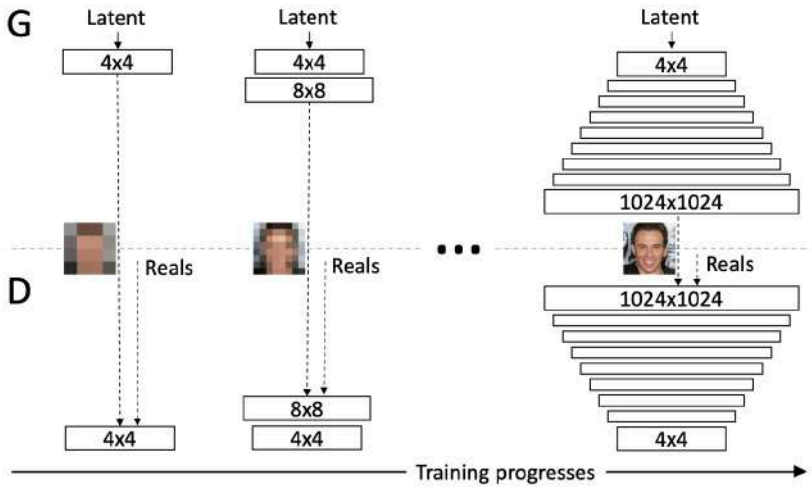

Figura 3.6: Arquitetura da ProGAN. Fonte: (KARRAS et al., 2018).

Entretanto, utilizar arquiteturas simétricas no gerador e discriminador não é via de regra. A SNGAN (MIYATO et al., 2018) propôs um discriminador mais estável, trazendo alterações nas camadas de D mas mantendo o gerador em uma arquitetura similar à DCGAN. A VAE-GAN (LARSEN et al., 2016) substitui a rede geradora para incluir uma rede Variational AutoEncoder. O resultado final será um método que combina a vantagem das GAN como um

modelo generativo de alta qualidade e das VAE como um método que produz um codificador de dados para o espaço latente.

A StyleGAN (KARRAS; LAINE; AILA, 2019) partiu da arquitetura da ProGAN, mas trouxe mudanças no gerador. Foi proposta a divisão do gerador em duas redes: uma *mapping network* f e uma *synthesis network* g. A rede de *synthesis* não recebe mais o vetor de ruído e começa com um ponto de partida constante aprendido. Quem recebe o vetor de ruído z é a rede de *mapping*, que tem como objetivo mapeá-lo para um espaço latente intermediário $\mathcal{W}$, $f : \mathcal{Z} \rightarrow \mathcal{W}$, produzindo w. Essa saída w é transformada nos vetores de estilo por meio de uma transformação paramétrica aprendida (blocos "A", na Figura 3.7) Esses vetores de estilo são passados a cada camada da rede de *synthesis*, controlando seu processo de geração por meio de adaptive instance normalization (AdaIN) (HUANG; BELONGIE, 2017).

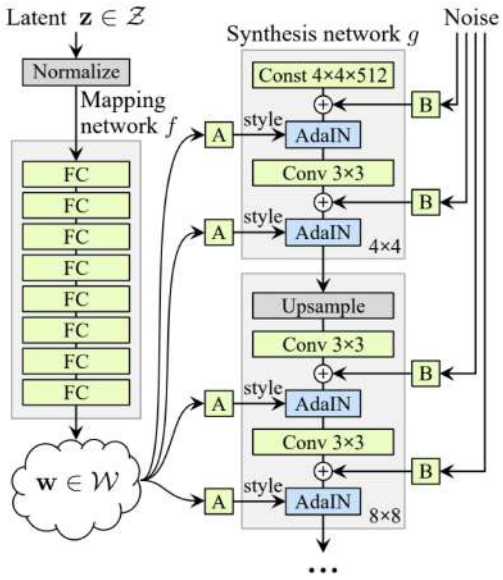

Figura 3.7: Arquitetura do gerador da StyleGAN. Fonte: (KARRAS; LAINE; AILA, 2019).

A aplicação dos vetores de estilo são incluídos a partir de convoluções moduladas. Essa modulação dimensiona cada feature map de entrada da convolução com base no estilo de entrada, o que pode ser alternativamente implementado escalando os pesos da convolução:

$$w'_{ijk} = s_i \cdot w_{ijk}$$

Onde w e w' são os pesos originais e modulados, respectivamente, s_i é a escala correspondente ao i-ésimo feature map de entrada, e j e k enumeram as feature maps de saída e a área espacial da convolução, respectivamente.

3.3.3 Treinamento e instabilidade

Treinar uma GAN consiste em encontrar o Equilíbrio de Nash em um jogo não-convexo com parâmetros contínuos em alta dimensão. Isso faz com que essa tarefa seja envolva em uma alta instabilidade. A atualização mútua do gerador e discriminador pode levar a uma série de problemas que impeçam qualquer uma das redes de convergir, fazendo com que os imagens obtidas não sejam realistas.

Uma característica relevante das GANs é que em todas as arquiteturas anteriores o gerador nunca observa as imagens reais, sendo guiado somente pelo gradiente da função de perda, o que faz com que o bom treinamento do discriminador seja essencial para a geração de imagens. Por outro lado, vale considerar que a tarefa de gerar imagens é um problema mais difícil do que a realizada pelos modelos discriminadores. Intuitivamente, é muito mais simples reconhecer uma pintura de Monet do que pintar como Monet. Assim, muito da dificuldade de treinamento é causada pela performance do gerador.

No contexto de geração de imagem, o *overfitting* da rede está fortemente relacionado à similaridade das imagens geradas com amostras dos dados reais. Uma métrica comumente utilizada para identificar esse *overfit* é encontrar o imagem vizinha real mais próxima (nearest neighbour) (THEIS; OORD; BETHGE, 2015).

Há também problemas relacionados à baixa variedade de geração. Nesse cenário, a rede é capaz de construir imagens, mas elas representam uma pequena parcela da da distribuição possível do conjunto de dados reais. Quando essa variância é pequena demais, a rede chega a um ponto chamado de *mode collapse*, no qual a rede geradora encontra um mínimo local que trava sua geração em um padrão específico de imagens. Dependendo do mínimo local, essas imagens podem inclusive não ser fidedignas.

Existem uma série de técnicas que buscam auxiliar na estabilidade da rede durante o treinamento da rede (SALIMANS et al., 2016). A adição de ruído também é uma técnica recorrentemente utilizada para o treinamento de redes neurais (AN, 1996), e também possui um impacto positivo em GANs (SØNDERBY et al., 2016). Além disso, a propensão de convergência da rede também está fortemente relacionada com os valores dos hiperparâmetros de treinamento da rede, como a taxa de aprendizagem, o tamanho das *mini-batches*, e o *tuning* dos otimizadores, como os β_1 e β_2 do otimizador Adam (KINGMA; BA, 2017).

Com o objetivo de evitar *overfitting* do discriminador, a proposta de *data augmentation* adaptativa foi proposta na StyleGAN2-ADA (KARRAS

et al., 2020), com o objetivo de aprimorar o treinamento de redes com um conjunto menor de dados reais. Em relação ao *overfit* do gerador, a proposta de *mini-batch discrimination* permite que o discriminador observe um conjunto de amostras geradas por vez. Já a ProGAN incorporou a proposta de *mini-batch standard deviation* que calcula valores estatísticos dos dados reais em uma camada ao fim do discriminador, que pode utilizar essas informações para discernir mini-batches geradas por sua baixa variação. Assim, incentivando a rede a gerar maior diversidade. Uma proposta de avaliação desse fenômeno é definir métricas de *recall*, também chamada de cobertura (*coverage*), que mede o quanto a distribuição aprendida pelo gerador cobre a distribuição real dos dados (SAJJADI et al., 2018).

3.4 Normalização de redes e SNGAN

O surgimento de redes neurais profundas trouxe um conjunto de novos desafios para a realização treinamentos efetivos. Abordagens de inicializações cuidadosas dos pesos da rede, com o monitoramento dos seus gradientes e ativações, se colocaram como essenciais para treinamentos que não divergissem (GLOROT; BENGIO, 2010).

Um dos fenômeno apontados como dificultadores no treinamento é a mudança da distribuição das entradas entre camadas da rede durante o treinamento, a chamada *internal covariance shift*. Resolver o *covariance shift* foi uma das principais motivações da criação da camada de Batch Normalization (IOFFE; SZEGEDY, 2015). Essa camada tem como objetivo normalizar cada feature independentemente, trazendo-as para uma média zero e variância unitária.

Assim, para uma camada com uma entrada $x = (x^{(1)}, ..., x^{(d)})$ de d dimensões, normaliza-se cada dimensão k com:

$$\hat{x}^k = \frac{x^{(k)} - E[x^{(k)}]}{\sqrt{Var[x^{(k)}] + \epsilon}}$$

No qual o valor esperado $E[x]$ e a variância $Var[x]$ são calculadas a partir de mini-batches do dataset de treino. É apontado que essa normalização acelera a convergência no treinamento mesmo quando essas dimensões são correlacionadas.

Essa normalização, porém, pode não ser suficiente para trazer estabilidade para o treinamento em uma GAN. Caso o discriminador alcance o chamado discriminador perfeito - sendo capaz de atribuir valor 0 a todas as imagens geradas como falsas e 1 a todas as verdadeiras - o gradiente dado ao

gerador passa a ter pouca ou nenhuma informação. Isso porque, caso o discriminador categorize perfeitamente, o gerador recebe valores muito semelhantes de perda para todas as imagens geradas, podendo levar a um *vanishing gradient*, quando os termos do gradiente se tornam extremamente baixos e não são suficientes para o treinamento da rede.

Tendo isso em mente, trabalhos posteriores argumentaram pela necessidade do discriminador respeitar a condição de continuidade de Lipschitz (ARJOVSKY; CHINTALA; BOTTOU, 2017; GULRAJANI et al., 2017).

Formalmente, dado uma função $f : \mathbb{R} \to \mathbb{R}$, dizemos que f é Lipschitz contínua se existe uma constante não negativa K tal que, para quaisquer dois números $x, y \in \mathbb{R}$, a seguinte condição é satisfeita:

$$|f(x) - f(y)| \leq K \cdot |x - y| \tag{3-4}$$

Ao limitar o gradiente do discriminador, se fornece mais informações no gradiente que ajudam a treinar o gerador. Procura-se, portanto, um discriminador a partir de um conjunto de funções contínuas K-Lipschitz:

$$\underset{||f||_{\text{Lip}} \leq K}{\text{argmax}} \, V(G, D)$$

Em que $||f||_{\text{Lip}}$ se refere à condição de Lipschitz apresentada na Equação 3-4. No geral, consideram-se funções que possuam a constante de Lipschitz $K = 1$, isto é, funções com uma inclinação limitada, que não são íngremes. Existem um conjunto de propostas para restringir o discriminador nesse sentido.

Na WGAN (ARJOVSKY; CHINTALA; BOTTOU, 2017), foi empregada a estratégia de limitar os pesos da rede em uma faixa fixa (por exemplo, $\mathcal{W} = [-0.01, 0.01]$) após cada atualização pelo gradiente. Entretanto, no próprio trabalho que a técnica foi introduzida foi apontado que isso pode facilmente levar a *vanishing gradients* quando o número de camadas aumenta.

Buscando superar essa limitação, o *gradient penalty* foi proposto na WGAN-GP(GULRAJANI et al., 2017). Uma função diferenciável é 1-Lipschitz se e somente se ela possui um gradiente de norma no máximo 1 em todos seus pontos. Assim, é adicionado na perda um termo que penaliza gradientes com norma maior que 1 da forma

$$\mathcal{L} = \mathbb{E}_{x \sim p_{\text{data}}}[f_\theta(x) - \lambda \, (||\nabla_{\hat{x}} D(\hat{x})||_2 - 1)^2] - \mathbb{E}_{z \sim p(z)}[f_\theta(G(x))]$$

Por fim, a normalização espectral (MIYATO et al., 2018) foi proposta. Ela limita a constante de Lipschitz em relação aos valores singulares de cada

camada $g : \mathbf{h}_{in} \rightarrow \mathbf{h}_{out}$ da rede ao limitar sua norma espectral da matriz de pesos W_l, na forma:

$$\bar{W}_{SN}(W) := W/\sigma(W)$$

Em que $\sigma(A)$ é a norma espectral da matriz A.

$$\sigma(A) = \max_{||\mathbf{h}||_2 \leq 1} ||A\mathbf{h}||_2$$

Assim, limita-se a dimensão da transformação realizada pelas camadas do discriminador, incentivando a estabilidade da rede. Em vez de se calcular a norma espectral a partir da decomposição em valores singulares, é estimado um valor $\sigma(A)$, o que traz um acréscimo bem pequeno de custo computacional no treinamento da GAN. Aplicando essa proposta, a Spectrally Normalized GAN (SNGAN) (MIYATO et al., 2018) foi proposta com uma arquitetura similar à DCGAN, mas com um discriminador que incorpora a normalização espectral em cada uma de suas camadadas. Além da inclusão dessa nova normalização, foram obtidos resultados superiores quando houve a retirada das camadas de Batch Normalization do discriminador, deixando-o somente normalizado a partir de sua norma espectral. A SNGAN atingiu resultados significativos em geração de imagem para uma série de datasets, como CIFAR-10, STL-10 e o ImageNet. Posteriormente, outros modelos, como a SAGAN (ZHANG et al., 2018) e a BigGAN (BROCK; DONAHUE; SIMONYAN, 2019), utilizaram normalização espectral tanto no discriminador quanto no gerador em busca da estabilidade de treinamento.

3.5 Dependência de longo alcance e self-attention

Capturar dependências de longo alcance é essencial em redes neurais profundas. Como apontado na seção 3.2, informações globais são adquiridas ao empilhar um conjunto de camadas convolucionais. Porém, depender de repetidas aplicações de convoluções que processam o entorno local é computacionalmente ineficiente e pode levar a dificuldades de otimização (WANG et al., 2018). Além disso, convoluções subsequentes levam a um impacto maior de pixels centrais da imagem no resultado final, dado seu maior número de possíveis ligações. Isso faz com que o campo receptivo efetivo seja consideravelmente menor do que o campo receptivo teórico (LUO et al., 2016), como mostra a Figura 3.8. É possível observar o formato gaussiano do campo receptivo efetivo. Nota-se também que ainda que o campo receptivo teórico seja maior do que o tamanho da imagem, o efetivo ainda não é capaz de capturar toda a imagem.

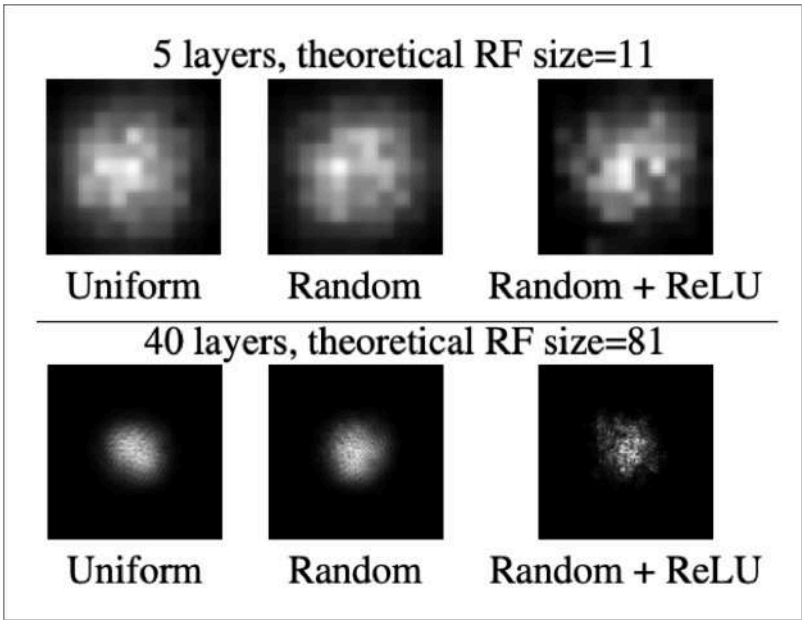

Figura 3.8: Comparação do efeito do número de camadas, inicialização de pesos aleatórios e função de ativação no campo receptivo efetivo em uma imagem 32×32. Fonte: (LUO et al., 2016)

Nesse contexto, a utilização de operadores não-locais pode trazer uma série de vantagems, como capturar a relação entre dois pontos de um *feature map* independente de suas posições e alcançar melhores resultados com o uso de menos camadas. Um operador não-local genérico (WANG et al., 2018) pode portanto ser representado pela equação

$$y_i = \frac{1}{\mathcal{C}(x)} \sum_{\forall j} f(x_i, x_j) g(x_j) \tag{3-5}$$

No qual y é o sinal de saída que deve ser calculado, x o sinal de entrada, e j é o índice que enumera todas as possíveis posições. Uma função par a par f computa o escalar que representa a relação entre i e j. A resposta é normalizada por um fator $\mathcal{C}(x)$. A não-localidade vem pelo fato de todas as posições $(\forall j)$ serem consideradas para o cálculo de y_i.

Um exemplo de operador não-local que obteve um impacto considerável no campo de visão computacional é o mecanismo de self-attention (VASWANI et al., 2017). Normalmente, self-attention é utilizado como um mecanismo de attention espacial - um método adaptativo de seleção de regiões, com o objetivo de responder *onde direcionar a atenção* afim de capturar informações globais.

Tendo uma imagem 2D como exemplo, dado um *feature map* $F \in C \times H \times W$, self-attention primeiro calcula as *queries*, chaves e valores $Q, K, V \in \mathbb{R}^{C' \times H \times W}$ por meio de projeções lineares e operações de remodelação. A self-attention pode então ser formulada como:

$$\begin{aligned} A = (a)_{i,j} &= \text{softmax}(QK^T) \\ Y &= AV \end{aligned} \tag{3-6}$$

Em que $A \in \mathbb{R}^{N \times N}$ é a matriz de atenção e $(a)_{i,j}$ é relação entre i e j. Pode-se observar que o self-attention é um caso especial da formulação genérica de um operador não local da Equação 3-5, em que para um dado i a expressão $\frac{1}{\mathcal{C}(x)} f(x_i, x_j)$ se torna a função softmax na dimensão j.

Essa formulação do mecanismo de atenção se demonstrou uma ferramenta poderosa de mapeamento de informações globais. Porém, ela possui algumas desvantagens, como um possível alto custo de memória e a complexidade quadrática que faz com que ela só possa ser utilizada após uma redução de dimensionalidade significativa (GUO et al., 2022; RAMACHANDRAN et al., 2019).

Dado seu sucesso, o mecanismo de atenção também foi incorporado no framework de GANs. A SAGAN, Self-Attention GAN (ZHANG et al., 2018), utiliza self-attention em diferentes estágios do gerador e do discriminador para modelar dependências de longo alcance. O trabalho também mostra que a aplicação do mecanismo de atenção em feature maps de nível médio a alto (mais próximos da imagem inicial ou final) entrega melhores resultados do que para feature maps de baixo nível (por exemplo, 4×4 e 8×8).

A camada de atenção recebe o feature map anterior x e, após uma redução de canal realizada por convoluções de kernel 1×1, calcula o mapa de atenção β_{ij}, representando o impacto da localização i ao gerar a região j na forma

$$\beta_{ij} = \text{softmax}(s_{ij}), \quad \text{onde } s_{ij} = f(x_i)^T g(x_j)$$

A partir de β_{ij}, a rede gera o mapa de autoatenção $o_i = \sum_{j=1}^{N} \beta_{i,j} \cdot V_j$, como pode ser observado na Figura 3.9. Esta saída é multiplicada por um parâmetro de escala aprendido γ e depois adicionada à entrada da seguinte maneira

$$y_i = \gamma o_i + x_i \tag{3-7}$$

O γ aprendido é inicialmente $\gamma = 0$ e permite que a rede gradualmente aprenda a atribuir mais peso às informações globais. Isso permite que o SAGAN se concentre primeiro nas informações locais mais simples e depois passe a aprender tarefas mais complexas. O modelo superou o estado-da-arte na época, e os pesquisadores observaram que a rede aprende padrões como semelhanças de cor e textura em locais distantes.

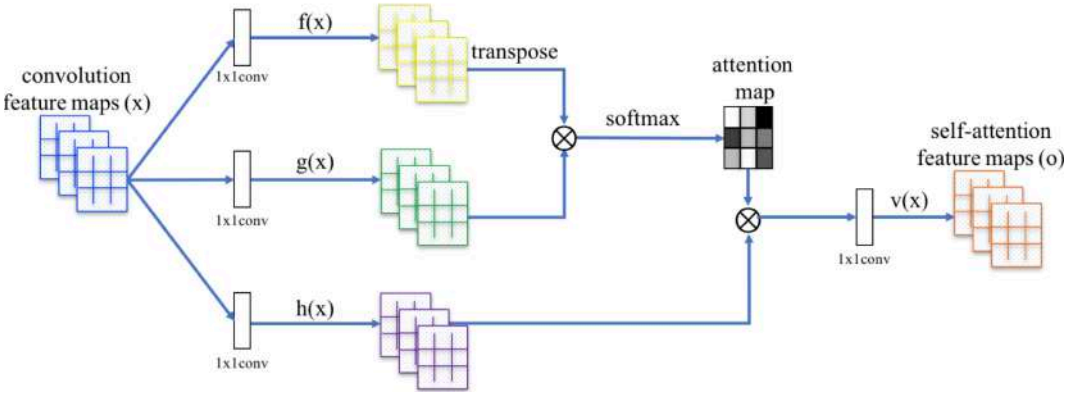

Figura 3.9: Arquitetura da camada de self-attention da SAGAN.

3.6 Fast Fourier Convolution

Aproveitando-se das propriedades teóricas da transformada espectral, a Convolução de Fourier Rápida, Fast Fourier Convolution (FFC) (CHI; JIANG; MU, 2020) foi concebida como um operador para alcançar campos receptivos globais por meio de convoluções no domínio de frequência. A FFC captura informações globais propagando dois sinais ao longo da profundidade da rede: um ramo local e um global. O *ramo local* x^l é formado por convoluções convencionais e espera-se que aprenda da vizinhança local $\mathcal{N}$. O *ramo global* x^g realiza operações no domínio de frequência, projetadas para capturar informações de longo alcance.

Formalmente, para uma entrada $\mathbf{X} \in \mathbb{R}^{H \times W \times C}$, a camada FFC divide os feature maps da entrada em $\mathbf{X} = \mathbf{X}^l, \mathbf{X}^g$. Essa divisão é controlada por um parâmetro $\alpha_{in} \in [0, 1]$, de modo que a dimensão de $\mathbf{X^l}$ e $\mathbf{X^g}$ seja, respectivamente, $H \times W \times ((1 - \alpha_{in}) \cdot C)$ e $H \times W \times (\alpha_{in} \cdot C)$.

Da mesma forma, o tensor de saída $\mathbf{Y}$ é dividido nos dois ramos $\mathbf{Y} = \mathbf{Y}^l, \mathbf{Y}^g$, e sua proporção é controlada por $\alpha_{out} \in [0, 1]$. Essa saída é formada da seguinte maneira:

$$\begin{aligned} \mathbf{Y}^l &= \mathbf{Y}^{l \to l} + \mathbf{Y}^{g \to l} = f_l(\mathbf{X}^l) + f_{g \to l}(\mathbf{X}^g) \\ \mathbf{Y}^g &= \mathbf{Y}^{g \to g} + \mathbf{Y}^{l \to g} = f_g(\mathbf{X}^g) + f_{l \to g}(\mathbf{X}^l) \end{aligned} \tag{3-8}$$

Onde f_l é uma convolução convencional, e $f_{l \to g}$ e $f_{g \to l}$ também são convoluções regulares que visam aproveitar os campos receptivos multi-escala. O termo f_g refere-se à Transformação Espectral, que realiza operações no domínio de frequência aplicando a Transformada Rápida de Fourier real e a Transformada Inversa de Fourier real. Em mais detalhes, o ramo global da FFC segue:

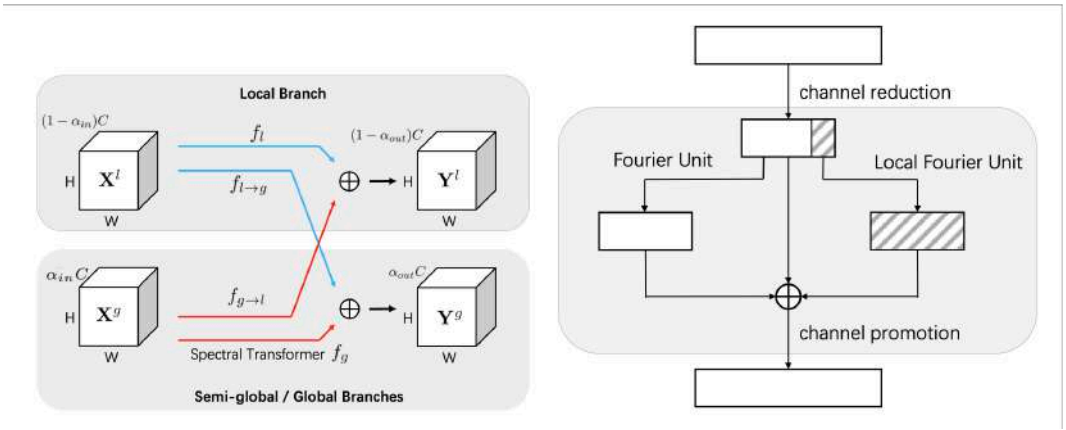

Figura 3.10: Arquitetura de um bloco de Fast Fourier Convolution.

- Executa uma convolução 1×1 na entrada f_g para realizar a redução de canais de entrada. Como descrito na proposta do módulo, os canais são reduzidos até metade de seu valor inicial (CHI; JIANG; MU, 2020).
- Aplica a Transformada Rápida de Fourier real e concatena as partes real e imaginária na dimensão de *features*.
- Aplica um bloco Conv-BN-ReLU com uma convolução 1×1 no domínio espectral. Como o teorema da convolução descreve, um tamanho de kernel maior não é necessário já que qualquer operação no domínio espectral possui um campo receptivo global.
- As *features* reais e imaginárias são separadas. Aplica a transformada inversa para retornar as informações ao domínio de pixels, entregando um sinal real de volta a rede.

Pode-se assim obter um campo receptivo que engloba toda a imagem com o módulo de Transformação Espectral da FFC, que calcula a transformada discreta de Fourier bidimensional real $\mathcal{F}(x) = X$ por canal. Na arquitetura completa de um bloco de Fast Fourier Convolution, também é proposta uma operação denominada Local Fourier Unit (LFU), que pode ser observado na Figura 3.10. A LFU foi projetada para capturar informações semi-globais e é usada em todos os blocos FFC. Porém, assim como descrito no modelo LaMa (SUVOROV et al., 2021), os blocos LFU não foram utilizados em nossa arquitetura.

Formalmente, seja um canal de imagem de entrada x de tamanho $N \times N$, e um único valor numérico desse canal $x[m, n]$, com largura $n \leq N$ e altura $m \leq N$. Considere uma onda senoidal bidimensional dada pelo produto

$$\begin{aligned} s_{k,l}[m, n] &= \omega_m \cdot \omega_n \\ \omega_m = e^{j2\pi \frac{k}{N} m} \quad &, \quad \omega_n = e^{j2\pi \frac{l}{N} n} \end{aligned} \tag{3-9}$$

Onde ω_m é a onda senoidal complexa vertical com frequência k/N e ω_n é a onda senoidal complexa horizontal com frequência l/N ciclos por amostra. A Transformada de Fourier decompõe os valores do canal em uma soma ponderada de ondas senoidais bidimensionais. A DFT2d é dada por

$$X[m,n] = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} x[k,l] \times e^{-j\Omega(km+ln)} \tag{3-10}$$

Onde $\Omega = 2\pi/N$ e X são os pesos da decomposição do domínio de frequência. A convolução no domínio espectral executa operações nessas ponderações e é capaz de ter uma interpretação em todo o canal da entrada em cada janela de convolução. Para reverter o sinal para o domínio de pixels, a DFT2d inversa é aplicada

$$x[m,n] = \frac{1}{N^2} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} X[k,l] \times e^{+j\Omega(km+ln)} \tag{3-11}$$

A intuição é que, ao aprender características no domínio de frequência, a rede é capaz de capturar dependências de longo alcance em todos os níveis de camadas. Além disso, uma atualização ponto a ponto das convoluções espectrais pode afetar a entrada globalmente, o que mostra seu potencial como um novo design arquitetural para campos receptivos não locais em GANs.

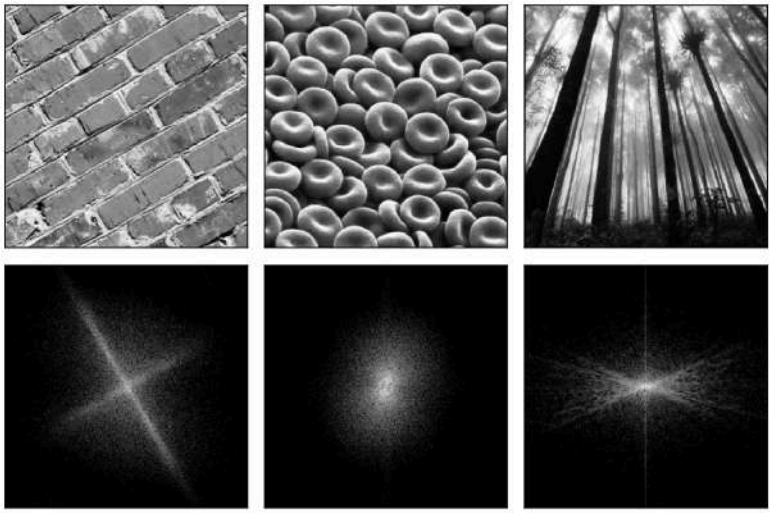

Figura 3.11: Exemplos de imagens de entrada e suas representações de frequência, apresentadas como log-amplitudes. Raios no domínio de frequência correspondem a bordas no domínio espacial alinhadas perpendicularmente a essas. Fonte: (RIPPEL; SNOEK; ADAMS, 2015)

As FFCs têm sido aplicadas com sucesso para obter resultados de ponta em tarefas de inpainting de imagens com máscara grande (SUVOROV et al., 2021; LU et al., 2022) e super-resolução (SINHA; MOORTHI; DHAR, 2022). Como discutido no trabalho SAGAN (ZHANG et al., 2018), convoluções re-

gulares têm dificuldade em capturar dependências de longo alcance e padrões geométricos e estruturais em classes de imagens. Levando isso em consideração, nosso trabalho levanta a questão de se o campo receptivo global das Convoluções de Fourier Rápidas pode beneficiar a síntese de imagens em um framework de Redes Generativas Adversárias.

4
Método proposto

Nesse capítulo é apresentada a arquitetura de rede proposta. Inicialmente, são detalhadas as mudanças realizadas para adequar os novos operadores de Fourier partindo da arquitetura tradicional da DCGAN. Em seguida, são levantadas os efeitos da inclusão das FFCs na rede, em especial na estabilidade e no custo computacional. Por fim, é apresentada a inclusão desses operadores em uma GAN mais complexa que obtém resultados estado-da-arte, a StyleGAN2.

4.1
Arquitetura Proposta: FCGAN

Como foi discutido na Seção 3.2, CNNs dependem fortemente do empilhamento de camadas convolucionais. Devido aos seus campos receptivos convencionais pequenos (3×3 ou 4×4), padrões estruturais só são visíveis para camadas superiores, tornando-as ineficientes para modelar dependências de longo alcance. Para lidar com esse problema, camadas de self-attention têm sido aplicadas à rede para capturar essas dependências. No entanto, quando confrontadas com *feature maps* pequenos (por exemplo, 8×8), camadas de self-attention desempenham um papel semelhante ao de convoluções locais (ZHANG et al., 2018). A self-attention também foi apontada como um fator de custo computacional elevado, sendo substituída com sucesso por módulos que utilizam Transformadas de Fourier para processamento de linguagem natural (LEE-THORP et al., 2021) e classificação de imagens (RAO et al., 2021).

O objetivo desse trabalho é aplicar operadores de Fast Fourier Convolution (FFC) em redes baseadas em convolução para introduzir campos receptivos globais em todos os níveis de *feature maps*. Chamamos essa rede de Rede Generativa Adversarial por Convolução de Fourier (FCGAN, do inglês Fourier Convolutional Generative Adversarial Network), devido à sua extensa aplicação de transformadas de Fourier em suas camadas. A arquitetura completa do FCGAN é apresentada na Figura 4.1. A intuição é que, ao aprender features no domínio da frequência, a rede é capaz de capturar dependências de longo alcance em todos os níveis das camadas. Além disso, uma atualização ponto a ponto das convoluções espectrais pode afetar globalmente a entrada, o que mostra seu potencial como um novo design para campos receptivos não locais em GANs. A FCGAN aplica operadores FFC em seu gerador, mantendo uma arquitetura mais tradicional no Discriminador.

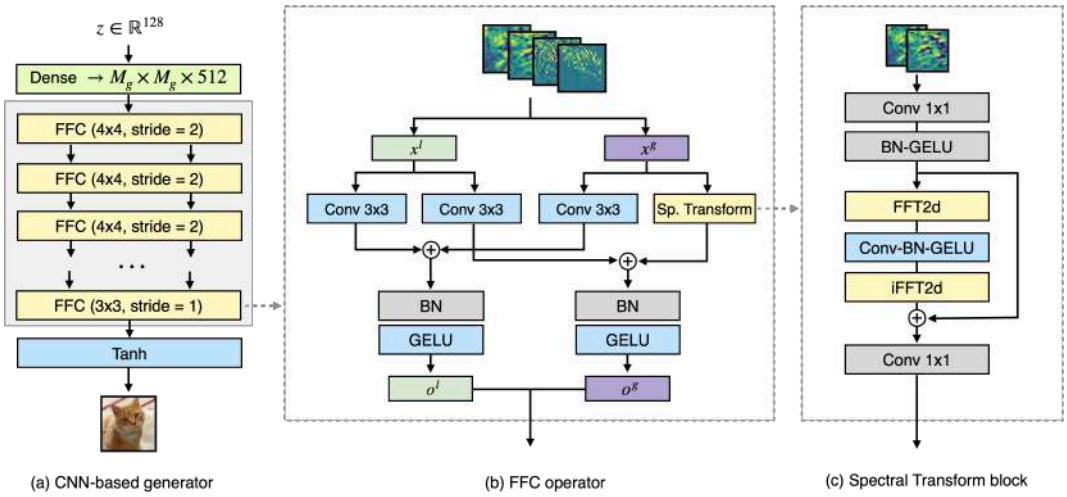

Figura 4.1: Arquitetura FCGAN Proposta.

A formulação da FCGAN teve como ponto de partida a arquitetura da DCGAN (RADFORD; METZ; CHINTALA, 2016). A DCGAN possui uma arquitetura simples que empilha blocos formados por uma camada de convolução, uma camada de batch normalization (BN) e uma ativação. Esse formato é seguido tanto no gerador quanto no discriminador, com a diferença da ativação ReLU em G, e LeakyReLU em D. A dimensionalidade espacial das imagens são dobradas no gerador e reduzidas pela metade no discriminador a cada bloco CONV-BN-ACT, como pode ser visto na Imagem 4.2. Inicialmente, foram substituídas todas as camadas de convoluções por camadas FFC, gerando assim blocos FFC-BN-ACT. Para a inclusão das FFCs em GANs estruturadas como ResNets, os blocos de G foram alterados conforme a Imagem 4.3.

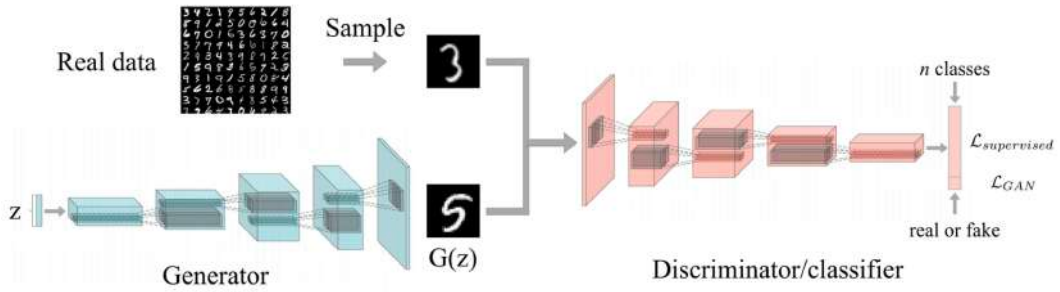

Figura 4.2: Arquitetura da DCGAN.

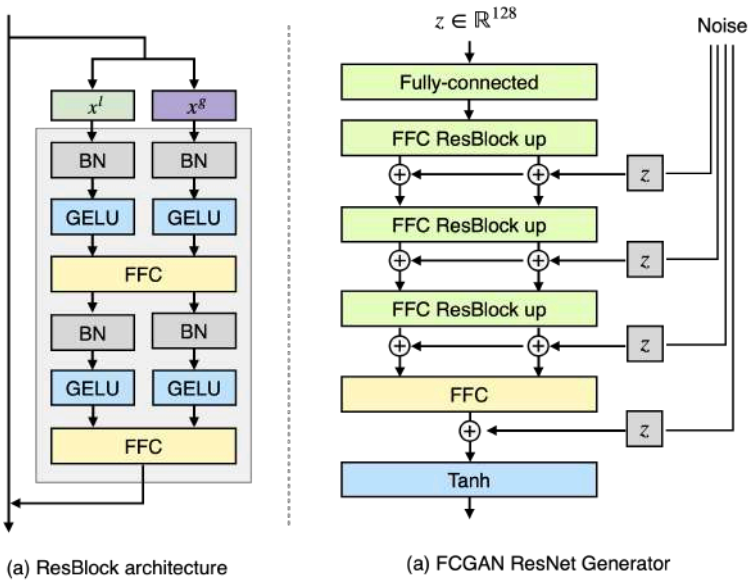

Figura 4.3: (a) Arquitetura da ResBlock. (b) Arquitetura da FCGAN baseada em uma ResNet.

4.2 Estabilização do treinamento

Com a adição do operador de Fast Fourier Convolution e sua escala mais ampla de manipulação, a instabilidade torna-se um desafio para o treinamento do modelo. Na arquitetura descrita na Seção 4.1, com tanto o discriminador quanto o gerador com camadas regulares de batch normalization, o treinamento é altamente instável e frequentemente diverge. Portanto, incorporamos camadas de normalização espectral como descrita na Seção 3.4 no modelo do discriminador a fim de restringir a constante de Lipschitz do modelo. Para o gerador, batch normalization é utilizado para o treinamento não-condicional e batch normalization condicional (CBN) para treinamento condicional (VRIES et al., 2017).

No gerador, a adição dos operadores FFC levou a uma ocorrência frequente de *mode collapse*. O gerador foi capaz de encontrar um pequeno conjunto de padrões que enganariam o discriminador, levando o treinamento à divergência. Esse problema foi resolvido com a injeção de ruído gaussiano como descrito na Seção 3.3.3 no final de cada bloco de convolução interno da FFC, conforme apresentado na Figura 4.3. O ruído adicionado está presente apenas durante o treinamento e impede que o gerador fique preso em mínimos locais. O ruído é adicionado tanto no ramo local quanto no global. Vale ressaltar que o ruído não é adicionado durante as operações no domínio da frequência, mas após a volta do sinal para o domínio espacial.

No seu artigo original, as FFCs são utilizadas no ImageNet para reco-

nhecimento de imagens, no Kinetics para reconhecimento de ações em vídeos e no MSCOCO para detecção de pontos-chave humanos (CHI; JIANG; MU, 2020). Isso poderia direcionar uma implementação que aproveita a FFC no discriminador. No entanto, ainda com o discriminador estabilizado da SNGAN, a substituição das convoluções tradicionais de D por operadores FFC levou à divergência ao treinamento. Uma explicação para esse fenômeno é que as operações do domínio da frequência não eram efetivamente normalizadas garantindo a condição de 1-Lipschitz. Ainda que a convolução espectral conforme com essa restrição, não é garantido que a operação resultante no domínio dos pixels esteja restrita a essa condição. O discriminador com operadores FFC, portanto, pode chegar mais facilmente ao discriminador ótimo e com isso causar fenômenos como o *vanishing gradient* (ARJOVSKY; CHINTALA; BOTTOU, 2017).

Além disso, trabalhos anteriores argumentaram a capacidade redes de detectar imagens geradas por meio das discrepâncias no espectro de Fourier. Redes que se atentam ao domínio da frequência são capazes de reconhecer com até 99% de acurácia imagens geradas por redes estado-da-arte, mesmo que estas sejam de difícil diferenciação por humanos (CHANDRASEGARAN; TRAN; CHEUNG, 2021b). Pode-se argumentar, portanto, que a adição de features no domínio da frequência ao discriminador pode levar à entrega de informações ao gradiente do gerador que não auxiliam na geração de imagens mais realistas. Desse modo, a arquitetura da FCGAN passou a ser composta por um discriminador conforme descrito na SNGAN, e um gerador no formato da DCGAN com os módulos FFC.

4.3 Features globais em um operador de menor custo.

Conforme discutido na Seção 3.5, modelar dependências de longo alcance por meio de self-attention possui um custo computacional alto. Trabalhos recentes sugerem o potencial operadores que extraem features no domínio de Fourier como substitutos de menor custo computacional para camadas de attention ou transformers (SUVOROV et al., 2021; LEE-THORP et al., 2021; RAO et al., 2021). Idealmente, aplicando a Transformada de Fourier - que não possui parâmetros treináveis - e tendo uma convolução no domínio da frequência, a rede será capaz de capturar dependências de longo alcance com menor custo computacional.

O operador FFC executa quatro convoluções em cada bloco da rede, como apresentado na Figura 4.1, em vez de apenas uma convolução convencional. No entanto, a criação do ramo global não impacta significativamente a contagem

de parâmetros ou o custo computacional. Como os canais são divididos entre os ramos local e global pelo hiperparâmetro α, a duração de cada etapa de atualização e o número de parâmetros treináveis são praticamente os mesmos de uma camada convolucional regular, como mostrado na Tabela 4.1. Em uma convolução de canais de entrada e saída C_1 e C_2, e um kernel $K \times K$, haveria $C_1 \times C_2 \times K^2$ parâmetros. Para um $\alpha_{in} = \alpha_{out} = 0.5$, a convolução local para local teria apenas um quarto dos parâmetros, ou seja, $0.5C_1 \times 0.5C_2 \times K^2$. Na FCGAN, utilizamos um valor de $K = 3$, assim como o descrito na proposta das camadas FFC. Para um $\alpha = 0.25$, o operador FFC tem no máximo 2% a mais de parâmetros do que uma convolução convencional (CHI; JIANG; MU, 2020).

Tabela 4.1: Informações sobre a contagem de parâmetros treináveis no FCGAN em uma arquitetura baseada em ResNet. Valores apresentados com um fator multiplicador de 10^{-6}. Os valores elevados para resolução de 48×48 são devido à decisão de arquitetura de uma camada densa inicial no SNGAN (MIYATO et al., 2018).

Modelo	**Tamanho da Imagem**	**Gerador**	**Discriminador**
SNGAN	32×32	1.15	1.05
SNGAN	48×48	4.87	10.14
SNGAN	64×64	3.88	7.13
FCGAN	32×32	1.16	1.05
FCGAN	48×48	4.93	10.14
FCGAN	64×64	3.84	7.13

Assim como é realizado na SAGAN, o custo para calcular a FFT2d e iFFT2d é controlado por uma convolução de kernel 1×1 que reduz os canais de entrada pela metade. Além disso, o FFC é versátil no sentido de que um $\alpha_{in}, \alpha_{out} = 0$ leva a uma operação de convolução regular. O único hiperparâmetro no FFC é α, que foi ajustado para $\alpha = 0.25$. Um valor de α maior (ou seja, $\alpha = 0.5$) não fez a rede divergir, mas reduziu as métricas obtidas nos experimentos.

4.3.1 Convoluções de Fourier em arquiteturas estado-da-arte

Buscando entender o potencial da aplicação de Convoluções Rápidas de Fourier em dimensões maiores, como 128×128 e 256×256, os blocos de FFC foram também incluídos em uma arquitetura da família StyleGAN. Como discutido na Seção 2.1, a StyleGAN já teve sucesso em implementar operações no domínio da frequência em sua versão StyleGAN3, com o foco de reduzir a geração de artefatos sintéticos na interpolação do espaço latente. Para

observar o resultado da adição dos operadores FFC, partiu-se da arquitetura da StyleGAN2, substituindo as convoluções moduladas da rede sintetizadora do gerador por operadores FFC modulados, como descritos na Imagem 4.4.

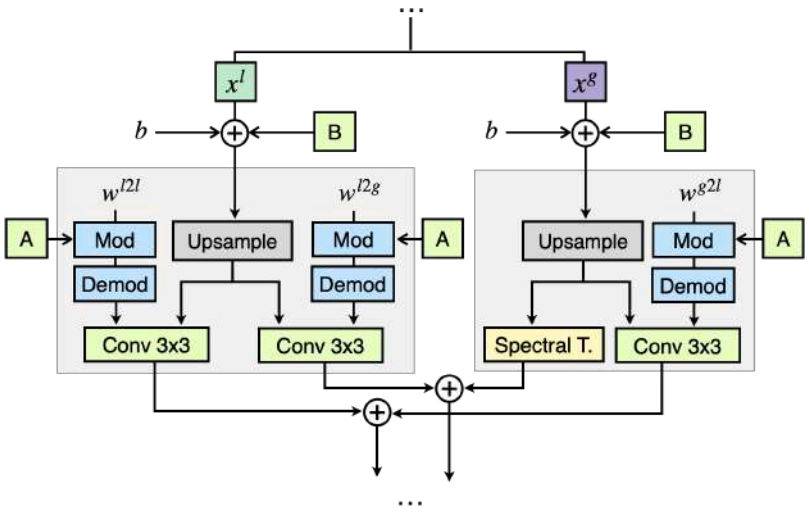

Figura 4.4: Arquitetura do bloco convolucional de FFC adaptado à StyleGAN2. Bloco A representa a inserção dos vetores de estilo, enquanto o bloco B aponta a inserção dos ruídos.

Dado o maior custo computacional da StyleGAN2, a Transformação Espectral foi alterada para melhor se adaptar às novas características da rede. Primeiramente, o ramo de LFU não foi utilizado. Além disso, foi adicionado um bloco de Squeeze and Excitation (SE) (HU et al., 2019) antes da convolução de kernel 1×1 com o objetivo de adaptar dinamicamente o peso das *feature maps* de entrada e permitir que as operações com Fourier fossem realizadas em canais mais informativos.

A partir dessa arquitetura, informações globais são propagadas durante toda a profundidade da rede. Os ventores de ruído são utilizados tanto no ramo local como no global, e os vetores de estilo são adicionados nas convoluções *local-to-global*, *local-to-local* e *global-to-local*. Os vetores de estilo e ruído são construídos do mesmo modo proposto pela StyleGAN2, e divididos entre os ramos locais e globais por meio da divisão dos canais dos tensores controlados pelo hiperparâmetro α.

5
Resultados

Nesse capítulo, são descritas as métricas e os conjuntos de dados utilizados para os experimentos na tarefa de geração de imagem. Em seguida, são apresentados os resultados quantitativos e qualitativos obtidos com o treinamento de duas arquiteturas de GANs que incluem blocos de Fast Fourier Convolutions, a FCGAN e a FCStyleGAN2.

5.1
Métricas de avaliação

Para avaliar os experimentos, optamos por calcular o Inception Score (IS) (SALIMANS et al., 2016) e a Fréchet Inception Distance (FID) (HEUSEL et al., 2018) como métricas quantitativas. Essas duas métricas são amplamente utilizadas para a tarefa de geração de imagem, e servirão de comparativo para trabalhos anteriores.

O Inception Score foi proposta com o objetivo de automatizar a avaliação das imagens geradas sem a necessidade de anotação humana. A métrica utiliza um classificador de imagens pré-treinando, comumente o modelo Inception. As imagens geradas são entregues ao modelo, que calcula a distribuição de rótulos $p(y|x)$. É esperado que uma imagem com objetos bem definidos tenha uma distribuição $p(y|x)$ com entropia baixa. A métrica é então calculada por:

$$\text{IS} = \exp\left(\mathbb{E}_x \mathbf{DKL}(p(y|x)||p(y)) \right)$$

Em que se calcula a divergência de Kullback-Leibler entre a distribuição $p(y|x)$ e a distribuição marginal $p(y)$, onde x é uma imagem gerada pelo gerador e y é o rótulo previsto pelo modelo Inception. Quanto maior o valor do IS, melhor o modelo em termos de qualidade e diversidade das imagens geradas.

O Inception Score, porém, possui algumas limitações. Devido ao fato de incorporar apenas as imagens geradas e não considerar imagens reais, o IS não consegue determinar a eficiência do gerador em modelos GANs. Além disso, ele não pode determinar se as imagens geradas estão bem alinhadas com a entrada fornecida ou não (BARAHEEM; LE; NGUYEN, 2023).

O Fréchet Inception Distance (FID) demonstrou ser mais consistente com a avaliação humana de realismo e variedade nas amostras geradas. Ele utiliza o modelo InceptionV3 para medir a distância entre as amostras geradas e as imagens reais, considerando assim as duas distribuições $p_g(x)$ e $p_{\text{data}}(x)$. O FID

é calculado a partir da média μ e covariância Σ das imagens reais r e geradas g conforme a equação:

$$\text{FID} = ||\mu_r - \mu_g||_2^2 + Tr(\Sigma_r + \Sigma_g - 2\sqrt{\Sigma_r \Sigma_g}) \tag{5-1}$$

Um FID menor indica uma distância mais curta entre as distribuições reais e as geradas. Assim, quanto menor o FID, melhor é a rede geradora. Porém, é importante ressaltar que trabalhos posteriores encontraram um forte viés no FID em relação à quantidade de amostras computadas n (BIŃKOWSKI et al., 2018). Assim, dois valores de FID só podem ser verdadeiramente comparados quando são calculados para a mesma quantidade de amostras. Ainda assim, não há garantia que o viés do FID será o mesmo ao comparar duas distribuições distintas.

Nos experimentos desse capítulo, medimos o FID usando todas as imagens de treinamento disponíveis e 10 mil amostras geradas. Para calcular o FID e o ISC, utilizamos o código disponível no pacote público *torch-fidelity* (OBUKHOV et al., 2020). Os modelos de referência para DCGAN e SNGAN foram obtidos a partir do *mimicry* (LEE; TRAN; CHEUNG, 2021).

5.2 Descrição das Bases de Dados

Uma rede generativa deve ter a capacidade de generalização para gerar imagens em conjuntos de dados grandes de alta variedade, ao mesmo tempo que não cair em *overfitting* ou *mode collapse* para conjuntos de dados menores. Assim, foram realizados experimentos em uma série de datasets de *benchmark*, são estes: CIFAR-10 (KRIZHEVSKY; HINTON et al., 2009), STL-10 (COATES; NG; LEE, 2011), SVHN (NETZER et al., 2011), Oxford Flowers 102 (NILSBACK; ZISSERMAN, 2008), CelebA (LIU et al., 2015), LSUN Bedroom (YU et al., 2015), e FFHQ (KARRAS; LAINE; AILA, 2019). A Tabela 5.1 entrega mais informações em relação aos conjuntos de dados. A Tabela 5.2 mostra um breve resumo da descrição das imagens presentes em cada *dataset*.

Dataset	n	**Condicionado**	**Classes**	**Resolução**
CIFAR-10	60.000	sim	10	32×32
STL-10	100.000	não	10	48×48
SVHN	600.000	sim	10	32×32
Oxford Flowers	8.189	não	102	128×128
CelebA	200.000	não	40	128×128
LSUN Bedroom	303.125	não	1	128×128
FFHQ	70.000	não	1	256×256

Tabela 5.1: Tabela de informações do datasets utilizados. O número de amostras n representa o total de imagens disponíveis para o treinamento. A coluna Condicionado informa se o treinamento realizado considerou ou não os atributos das classe. A coluna Resolução explicita a resolução máxima de treinamento realizado com cada conjunto de dados.

Dataset	Descrição
CIFAR-10	Imagens em baixa resolução de 10 classes: avião, automóvel, pássaro, gato, veado, cachorro, sapo, cavalo, navio e caminhão.
STL-10	Imagens distribuídas em 10 classes: avião, pássaro, carro, gato, veado, cachorro, cavalo, macaco, navio, caminhão. A maioria dos dados é não rotulado.
SVHN	Imagens de placas de números de casas com múltiplos algarismos entre 0 e 9 retiradas do Google Street View.
Oxford Flowers	Imagens de 102 categorias de flores. Cada classe consiste em entre 40 e 258 imagens.
CelebA	Imagens de rostos de mais de 100 mil celebridades. Imagens cobrem uma grande variedade de poses e fundos.
LSUN Bedroom	Subconjunto do dataset LSUN contendo somente imagens de quartos (cômodos) contendo objetos recorrentes como camas, almofadas, janelas e luminárias.
FFHQ	Imagens de rostos em alta resolução contendo uma variedade considerável de idade, etnicidade e fundo. Possui também uma grande variedade de acessórios, como óculos, óculos escuros, chapéus, entre outros.

Tabela 5.2: Tabela com a descrição dos itens presentes nos conjuntos de dados de treinamento.

5.3 Implementação e Hiperparâmetros

A implementação dos modelos da rede FCGAN e FCStyleGAN foram desenvolvidas na linguagem Python por meio do framework PyTorch [1]. Os módulos de Fast Fourier Convolutions foram incorporados a partir do código fonte oficial disponibilizados pelos pesquisadores [2]. A implementação da SNGAN e DCGAN seguiram as arquiteturas divulgadas em seus respectivos artigos. A implementação da StyleGAN2 utilizada foi disponibilizada em uma biblioteca open source [3].

Por padrão, o treinamento do FCGAN para dimensões de 32×32 e 64×64 foi realizado em uma única GPU NVIDIA T4 em 100.000 passos e um *batch size* de 64. Inspirado nas atualizações no design do ResNet apresentadas por (LIU et al., 2022), usamos o otimizador AdamW (LOSHCHILOV; HUTTER, 2019) em vez do Adam convencional. Os hiperparâmetros utilizados estão detalhados na Tabela 5.3.

Tabela 5.3: Hiperparâmetros utilizados nos treinamentos

Modelo	**Otimizador**	β_1	β_2	**GPU**
FCGAN (CNN)	AdamW	0.5	0.999	NVIDIA T4
FCGAN (ResNet)	AdamW	0.0	0.9	NVIDIA T4
FCStyleGAN	Adam	0.0	0.9	NVIDIA V100

Além disso, para os modelos FCGAN, a função de ativação GELU (Gaussian Error Linear Units) (HENDRYCKS; GIMPEL, 2020) foi usada no gerador em vez de ReLU, enquanto o discriminador usou LeakyReLU com uma inclinação negativa de 0.1. Para a FCStyleGAN e resoluções $128 \times 128 e 256 \times 256$, o treinamento foi realizado em uma única GPU NVIDIA V100, em um *batch size* de 32 para imagens 128×128. Para a resolução 256×256, foi utilizado um *batch size* de 12 amostras com 4 acumulações de gradiente.

Devido a um alto custo computacional e de tempo atrelado ao treinamento dos modelos, não foram realizados métodos para ajustes finos de hiperparâmetros. Os hiperparâmetros utilizados são, portanto, os descritos nos trabalhos originais das redes *baseline*.

Todos os treinamentos foram realizados em uma infraestrutura em nuvem, por meio da plataforma Google Colab [4]. Nele, são disponibilizados três possiblidades de placa de vídeo: NVIDIA T4, NVIDIA V100 e NVIDIA A100. Dado o custo financeiro da utilização dessas placas, foi utilizada a NVIDIA

[1] https://pytorch.org/
[2] https://github.com/pkumivision/FFC
[3] https://github.com/lucidrains/stylegan2-pytorch
[4] https://colab.research.google.com

T4 para dimensões mais baixas (32×32 e 48×48) e a NVIDIA V100 para dimensões mais altas (128×128 e 256×256).

A FCGAN e a FCStyleGAN são treinadas para minimizar a *hinge loss* adversarial (LIM; YE, 2017) tanto para o discriminador quanto para o gerador. Essa abordagem foi testada tanto no SAGAN (ZHANG et al., 2018) quanto no SNGAN (MIYATO et al., 2018), substituindo a função de perda convencional proposta em (GOODFELLOW et al., 2014). A *hinge loss* para o discriminador e o gerador é dada respectivamente por:

$$\begin{aligned} L_D = & - \mathbb{E}_{(x,y)\sim p_{data}}[\, \min(0, -1 + D(x,y))\,] \\ & - \mathbb{E}_{z\sim p_z, y\sim p_{data}}[\, \min(0, -1 - D(G(z), y\,] \\ L_G = & - \mathbb{E}_{z\sim p_z, y\sim p_{data}}\, D(G(z), y) \end{aligned} \tag{5-2}$$

5.4 Descrição dos experimentos quantitativos na FCGAN

Para entender e avaliar o impacto da adição de convoluções de abrangência global em um framework GAN, realizamos experimentos com os conjuntos de dados CIFAR-10, SVHN e STL-10. A escolha por esses conjuntos de dados para a avaliação quantitativa se dá devido ao alto custo de tempo e hardware necessário para computar valores de FID e ISC durante o treinamento. Assim, experimentos quantitativos com CIFAR-10, SVHN e STL-10 se tornam mais acessíveis por estes serem formados por imagens de menor resolução (respectivamente, 32×32, 32×32 e 48×48). Nesta subseção, relatamos o desempenho do nosso modelo e as implicações de um conjunto de decisões de design na eficácia da rede.

A Tabela 5.4 mostra detalhes dos experimentos com CIFAR-10. Realizamos um estudo detalhado para entender os impactos individuais de cada novo componente adicionado ao modelo. Como detalhado no capítulo 4, começamos implementando blocos FFC no lugar de convoluções convencionais em uma arquitetura semelhante à DCGAN (RADFORD; METZ; CHINTALA, 2016). As features de Fourier adicionadas não foram suficientes para superar os resultados do SNGAN e levaram a um treinamento instável. Implementando os métodos discutidos anteriormente na Seção 4.2, conseguimos estabilizar o treinamento e obter um desempenho ligeiramente melhor do que o SNGAN convencional. Com uma arquitetura baseada em ResNet, a FCGAN alcançou um FID de $18,98$ e um Inception Score de $8,14$ - uma redução de $4,98$ no FID.

A FCGAN também obteve melhores resultados no treinamento incondicional com o STL-10. A Tabela 5.6 e a Tabela 5.5 apresenta um resumo dos resultados quantitativos de nosso modelo em termos de FID e ISC, respectiva-

mente. A FCGAN superou os modelos de referência tanto no Inception Score quanto no FID, para o mesmo número máximo de 100.000 iterações. Para o DCGAN, as métricas de FID e IS dos conjuntos de dados CIFAR-10 e STL-10 foram obtidas de (HEUSEL et al., 2018; HE et al., 2019) e (NGUYEN et al., 2017).

Tabela 5.4: Estudos detalhados das adições à rede para a construção da FCGAN. Resultados obtidos com o dataset CIFAR-10.

Modelo	FID	IS
SNGAN	23.9	7.97
+ Self-Attention no Generator	23.7	7.45
+ Self-Attention em D e G	47.7	6.40
FFC em G, Discriminator DCGAN	24.5	7.05
+ FFC em D e G	45.9	5.73
FFC em G, Discriminator SNGAN	23.2	7.63
+ Conditional Batch Normalization	22.9	7.54
+ GELU and AdamW	21.4	7.80
+ Arquitetura ResNet	**18.98**	**8.14**

As métricas para o SNGAN foram obtidas de nossos experimentos executando o código disponível na biblioteca *mimicry* (LEE; TRAN; CHEUNG, 2021). No artigo original, os experimentos do SNGAN apresentaram valores mais altos de Inception Score, 8.22 ± .05 e 9.10 ± .04, respectivamente, para CIFAR-10 e STL-10. Ao tentar recriar esses resultados, (LEE; TRAN; CHEUNG, 2021) obteve uma IS de $7,97$ no CIFAR-10 e uma IS de $8,04$ no STL-10. Levando isso em consideração, optamos por relatar os números alcançados por nossos experimentos, garantindo as mesmas configurações de avaliação e treinamento para SNGAN e FCGAN.

Tabela 5.5: Resultados quantitativos da FCGAN no CIFAR-10 e no STL-10.

	Inception Score ↑		
Método	CIFAR-10	STL-10	SVHN
Dados Reais	11.24	26.08	3.31
DCGAN	6.64	7.54	2.99
SNGAN	7.97	8.48	3.04
FCGAN	8.14	8.84	3.09

Tabela 5.6: Resultados quantitativos da FCGAN no CIFAR-10 e no STL-10.

Método	FID ↓ CIFAR-10	STL-10	SVHN
Dados Reais	7.80	7.90	1.15
DCGAN	28.95	51.01	33.09
SNGAN	23.90	40.10	13.32
FCGAN	18.98	38.71	12.77

5.5 Descrição dos experimentos qualitativos na FCGAN

Para realizar a análise qualitativa do desempenho da rede, conduzimos uma avaliação em três conjuntos de testes de características distintas: o SVHN, Oxford Flowers e o CelebA. O objetivo dessa avaliação é verificar a qualidade das imagens geradas além das métricas quantitativas, em uma variedade de cenários e tipos de dados.

5.5.1 SVHN

O treinamento do SVHN foi realizado de maneira condicionada, ainda que o conjunto de dados apresente o desafio de poder conter mais de um dígito por imagem. Além disso, comparando-o com outros conjuntos de dados de algarismos, como o MNIST (LECUN; CORTES; BURGES, 2010), o SVHN possui três canais de cor e uma variedade consideravelmente maior de formatos, fundos, iluminação e resolução. Na figura 5.1, é possível comparar as amostras geradas e selecionadas do FCGAN com os dados reais. As imagens foram geradas na resolução de 32×32 (tamanho original do dataset) com um *batch size* de 64 e 100.000 passos de treinamento do gerador, usando os mesmos hiperparâmetros do treinamento com o CIFAR-10.

Pode-se observar que a rede é capaz de compreender padrões distintos dentro da mesma classe de dígitos. Algumas das imagens geradas possuem mais de um algarismo, assim como o observado nas imagens reais. A rede, porém, apresenta dificuldade em equiparar a variedade dos dados reais, apresentando em sua maioria exemplos de cores azul e branco.

5.5.2 Oxford Flowers

A geração a partir do dataset Oxford Flowers traz o desafio de um conjunto de dados menor. Como pode ser observado na Tabela 5.1, o número de imagens reais disponíveis para treinamento é consideravelmente inferior aos

Figura 5.1: Números gerados (esquerda) por meio da FCGAN com o dataset SVHN (direita).

outros datasets. Além disso, o maior número de classes presentes entrega uma maior diversidade com um número menor de exemplos. Isso torna os dados disponíveis insuficientes para a rede aprender as features necessárias para gerar novas imagens.

Portanto, foi realizado uma rotina de *data augmentation* para aumentar o conjunto de imagens para 32.756 (oito vezes a quantidade original). Foram realizadas inversões verticais e horizontais, recortes e rotações aleatórias e transformações de cores para aumentar o número de imagens disponíveis.

A Imagem 5.2 traz exemplos sintetizados de tamanho 48×48 em 80 mil, 90 mil e 100 mil passos do gerador. A FCGAN foi treinado com um *batch size* de 64, $\alpha = 0.25$, e uma taxa de aprendizado de 0.0002. Mesmo em um conjunto de dados menor (com menos de 40 mil imagens), a FCGAN é capaz de capturar padrões estruturais significativos das classes altamente geométricas, gerando um conjunto diversificado de tipos de imagens e flores no treinamento incondicional.

Pode-se observar que algumas imagens na Imagem 5.2 possuem resquícios da rotina de *data augmentation*. Esse é um problema comum no treinamento de GANs com poucas imagens muitas vezes gerado pelo *overfit* do discriminador. No trabalho da StyleGAN2-ADA (KARRAS et al., 2020), foi proposto o mecanismo de *adaptive discriminator augmentation*, que estabiliza o treinamento para datasets pequenos. Entretanto, a inclusão desse mecanismo no discriminador seria uma mudança considerável na arquitetura da FCGAN. A

Figura 5.2: Amostras 48 × 48 selecionadas do treinamento do FCGAN no conjunto de dados Oxford Flowers.

FCStyleGAN se aproxima dessa arquitetura, e seus resultados são descritos na Seção 5.8.

Buscando observar a robustez da rede para capturar padrões globais em maiores resoluções, foi realizado o treinamento da FCGAN para resolução 128 × 128. O resultado do treinamento pode ser observado na Imagem 5.3.

Figura 5.3: Flores geradas na resolução 128×128 pela FCGAN.

5.5.3 CelebA

A sintetização de rostos humanos é uma tarefa comum de benchmark para modelos de geradores de imagem. Por estarmos acostumados a ver rostos

cotidianamente, detalhes mal-acabados ou que não carregam uma consistência gera estranheza. Além disso, rostos humanos possuem uma série de *features* que devem ser compreendidas para que a geração de imagem seja efetiva.

Buscando observar o desempenho da FCGAN nessa tarefa, foi realizado o treinamento de um modelo com o dataset CelebA (Large-scale CelebFaces Attributes). Foram treinados modelos para três dimensões distintas: 48×48, 64×64 e 128×128. O treinamento realizado foi não condicionado, tendo o discriminador somente acesso às imagens reais e geradas, e o gerador ao resultado da função de perda.

Figura 5.4: Rostos gerados a partir do dataset CelebA nas resoluções 48×48, 64×64 e 128×128.

A Figura 5.4 apresenta amostras selecionadas da FCGAN. Os treinamentos foram realizados com um *batch size* de 64 e uma taxa de aprendizagem de 0.0001. Para o treinamento em maior resolução (128×128), foram adicionadas camadas ao gerador e discriminador de maneira análoga à transformação do modelo da dimensão 32×32 para 64×64. Nenhuma outra técnica foi utilizada para estabilizar o treinamento ou melhorar a qualidade das imagens geradas.

A rede FCGAN foi capaz de gerar imagens realistas em todas as resoluções de treinamento. Os rostos gerados variam em idade, gênero e raça, além de elementos como cores de cabelo e barba. Os resultados qualitativos apresentados na Figura 5.4 também demonstram o potencial das convoluções rápidas de Fourier em resoluções maiores, resultado também encontrado em trabalhos anteriores (SUVOROV et al., 2021).

5.6 Padrões estruturais e Features em Fourier

Com o objetivo de observar o impacto das operações de Fourier na geração das imagens, foram observados os *feature maps* do treinamento com o dataset Oxford Flowers e CelebA.

Imagens de flores possuem comumente padrões geométricos ao longo da imagem como um todo, principalmente no formato de suas pétalas. As contribuições dos ramos local e global dos operadores FFC podem ser observadas na Figura 5.5. É possível observar que o ramo local e global constroem partes diferentes da imagem. A última convolução da rede recebe os os *feature maps* do ramo local e global concatenados e entrega a imagem em três canais de cor.

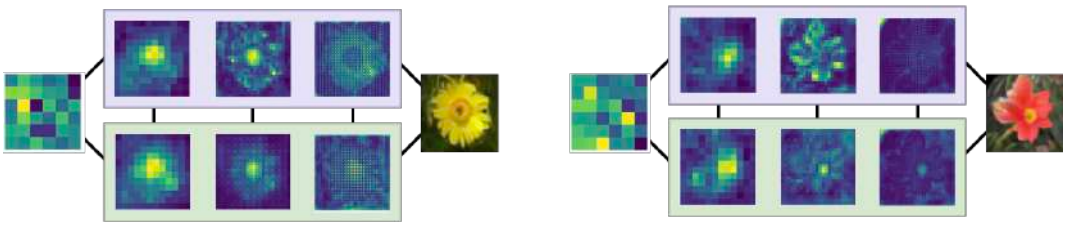

Figura 5.5: Feature maps das convoluções da FCGAN para amostras do dataset Oxford Flowers. Em verde, o ramo local da FFC. Em roxo, o ramo global.

Vale reforçar que o cálculo dos *feature maps* locais levam em consideração o resultado do ramo global da camada anterior, como é mostrado na Equação 3-8. Assim, as contribuições dos ramos locais e globais são combinadas desde as primeiras camadas da rede.

A Figura 5.6 apresenta dois exemplos dos *feature maps* locais e globais para a geração de rostos pela FCGAN. Assim como na geração das flores, cada um dos ramos entrega um foco maior em partes distintas da imagem gerada. No caso dos rostos, ambos ramos são capazes de identificar a área dos olhos e sorriso, além de gerar detalhamentos consideráveis nos cabelos.

Ainda que o ramo global com *features* de Fourier não atrapalhasse a geração de imagens, um fenômeno possível seria a não-utilização desse ramo pela rede. Isto é, na otimização dos parâmetros, as operações com Fourier serem reduzidas ao ponto de não entregar informações relevantes para a síntese das imagens. Ao observar os *feature maps* do ramo global, percebemos que eles possuem informações relevantes para as camadas futuras que são utilizadas para gerarem imagens mais realistas, como mostram os resultados qualitativos e quantitativos desse capítulo.

Como apontado na Seção 3.5, os campos receptivos efetivos de redes convolucionais são menores do que os teóricos, apresentando um comportamento

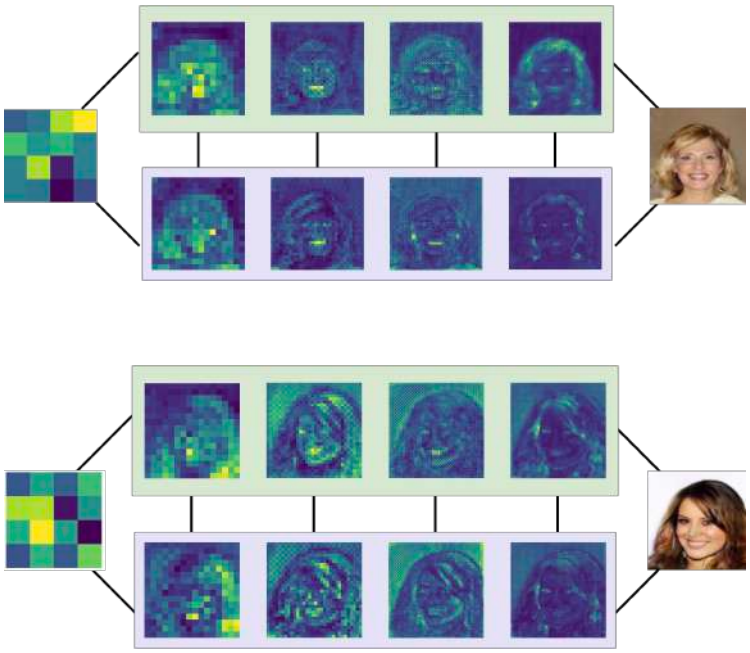

Figura 5.6: Feature maps das convoluções da FCGAN para amostra do dataset CelebA. Em verde, o ramo local da FFC. Em roxo, o ramo global.

próximo de gaussiano. Na teoria, as Convoluções Rápidas de Fourier possuem um campo receptivo de abrangência global. Para observar esse comportamento, realizou-se o mesmo experimento descrito em (LUO et al., 2016).

Dado que os pixels de cada camada são indexados por (i, j), com o centro em $(0, 0)$. Considere o (i, j)-ésimo pixels da p-ésima camada $x^p_{i,j}$, com $x^0_{i,j}$ representando a entrada da rede e $y_{i,j} = x^n_{i,j}$ a saída na n-ésima camada. O campo receptivo efetivo (ERF) é definido como a região de entrada que possua qualquer impacto não-insignificante em $y_{0,0}$. O impacto de um pixel de entrada $x^0_{i,j}$ pode ser calculado pela derivada parcial $\partial y_{0,0}/\partial x^0_{i,j}$. Por meio de back-propagation, podemos calcular essa derivada parcial definindo o gradiente de erro como $\partial l/\partial y_{0,0} = 1$ e $\partial l/\partial y_{i,j} = 0$ para todo $i \neq 0$ e $j \neq 0$.

Assim como em (LUO et al., 2016), pesos das convoluções foram inicializados aleatoriamente e a função de ativação não-linear ReLU foi utilizada na saída de cada camada. Como o impacto em $y_{0,0}$ é dependente da entrada e dos pesos aleatorizados, os resultados foram obtidos pelo impacto médio após 100 execuções. A Figura 5.7 compara o ERF de uma rede convolucional tradicional e uma rede convolucional com módulos FFC de $\alpha = 1.0$.

Na primeira linha (a), é mostrado o resultado de uma rede convolucional com 5 (coluna um) e 50 camadas (coluna dois). Cada camada é composta por uma convolução de kernel 3×3 e a ativação ReLU. Observa-se que com essa arquitetura o campo receptivo efetivo tem formato quase gaussiano, assim como o descrito por (LUO et al., 2016). Além disso, o ERF atinge apenas

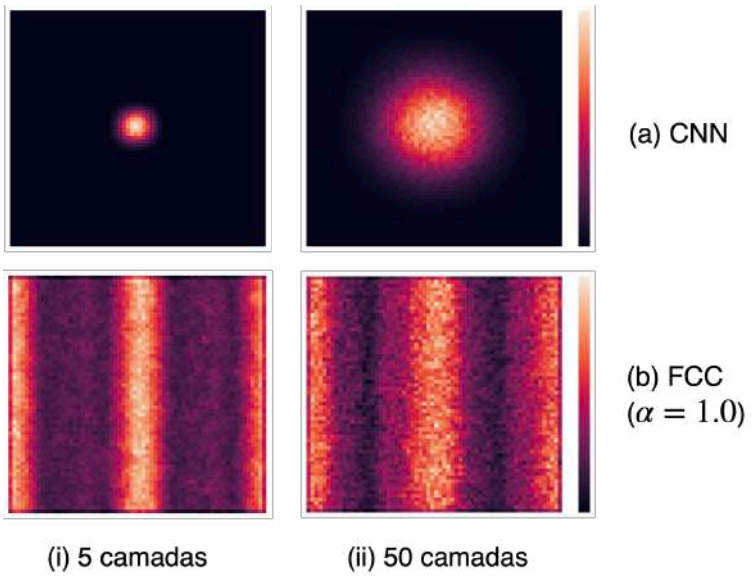

Figura 5.7: Comparação do campo receptivo efetivo (ERF) de uma CNN e uma rede convolucional com módulos FFC em uma entrada 64×64.

parte da imagem em 50 camadas, ainda que o campo receptivo teórico (TRF) tenha tamanho maior que a imagem, TRF= 101. Em contraste, a rede com convoluções espectrais entrega uma ERF que se expande por todo o canal da imagem, permitindo que cada pixel de entrada $x^0_{i,j}$ impacte $y_{0,0}$. Esse comportamento é alcançado desde as primeiras camadas da rede, o que entrega uma maior capacidade de captação de padrões estruturais.

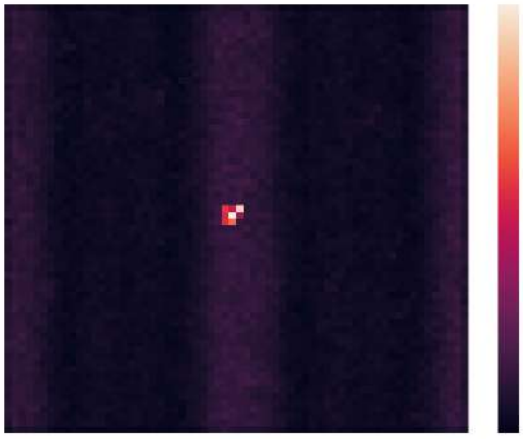

Figura 5.8: Visualização da contribuição dos ramos globais e locais da FFC no feature map de saída do bloco.

Na Figura 5.8, é possível também observar a participação dos dois ramos ao calcular um feature map de saída. Em uma variação da rede convolucional anterior, definimos $\alpha = 0.5$. Com esse valor, temos uma distribuição igualitária entre os canais de entrada para o ramo local e para o ramo global. Pode-se observar na Figura 5.8 a contribuição da informação global em um formato

similar a uma senóide, enquanto um quadrado de tamanho 3×3 (mesmo tamanho do kernel das convoluções) é formado com mais destaque.

5.7 Comparações com Self-Attention

Para avaliar a influência no custo computacional de um operador FFC em comparação com uma camada de self-attention, treinamos uma alteração do modelo SNGAN no CIFAR-10 com as mesmas especificações do treinamento da SNGAN e da FCGAN. Adicionamos camadas de attention na mesma posição das camadas relatadas no discriminador e gerador do SAGAN, priorizando os *feature maps* de alta dimensionalidade.

A adição de self-attention levou a um treinamento que é 39.8% mais lento do que a SNGAN regular. Em comparação, o treinamento do FCGAN leva apenas 12.1% a mais do que o da SNGAN. Treinando nas mesmas condições de hardware, adicionar uma única camada de self-attention no discriminador e gerador é 24.7% mais lento do que aplicar blocos FFC em todo o gerador do FCGAN, como pode ser observado na Figura 5.11.

Além disso, a FCGAN não exigiu camadas adicionais de normalização no gerador para a estabilização do treinamento. A SAGAN adiciona a normalização espectral ao gerador juntamente com camadas de Batch Normalization para estabilizar o treinamento. Simplesmente adicionar as camadas de self-attention do SAGAN à SNGAN resulta em métricas ruins, com um FID de 47.7 e um valor de Inception Score de 6.40. Ao empregar self-attention apenas no gerador, em uma configuração semelhante à do FCGAN, conseguimos um FID de 23.7 e um ISC de 7.45, que são resultados inferiores aos observados no FCGAN com arquitetura semelhante, conforme relatado na Tabela 5.4.

As métricas mais baixas alcançadas neste experimento indicam um comportamento menos eficaz da attention em *feature maps* pequenos. Ao mesmo tempo, aponta para um alto custo computacional quando aplicadas para altas dimensões de imagens. Em resumo, esses resultados ajudam a demonstrar o potencial das features de Fourier como uma abordagem geral de menor custo para capturar informações globais.

A Figura 5.9 e a Figura 5.10 mostram a comparação entre a SNGAN e a FCGAN para valores diferentes do hiperparâmetro α. Foram comparados $\alpha = 0.25$ e $\alpha = 0.50$, valores utilizados no trabalho original (CHI; JIANG; MU, 2020). Com o valor de α menor, a FCGAN consegue alcançar um melhor FID. Nessa configuração, a FCGAN também é mais eficiente computacionalmente, levando 9% menos tempo em 10 mil passos, como mostra a Figura 5.11. A FCGAN é $12,1\%$ mais lento que a SNGAN, mas $24,7\%$ mais rápido que a

self-attention. Para os resultados listados acima, utilizou-se uma única GPU NVIDIA T4.

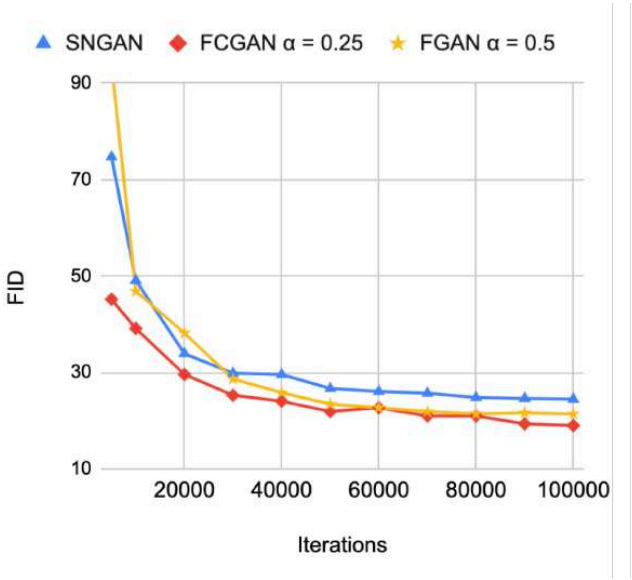

Figura 5.9: Gráfico de FID no CIFAR-10. Comparação entre SNGAN e FCGAN com $\alpha = 0,25$ e $\alpha = 0,50$.

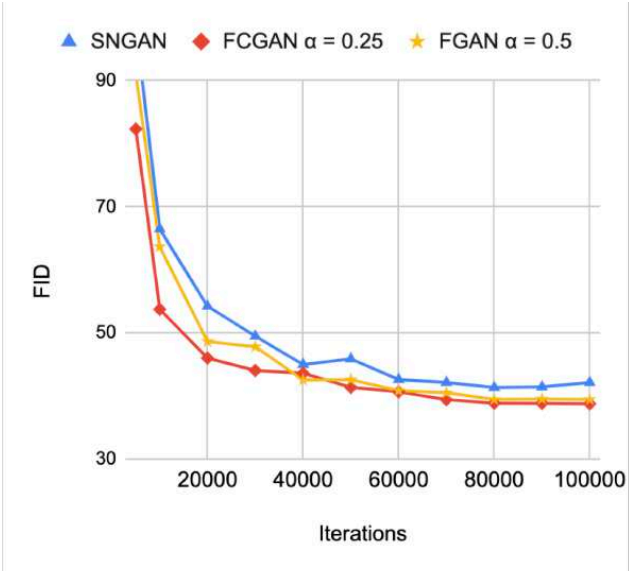

Figura 5.10: Gráfico de FID no STL-10. Comparação entre SNGAN e FCGAN com $\alpha = 0,25$ e $\alpha = 0,50$.

Para observar o impacto do uso de módulos FFC em dimensões mais altas, explorou-se o custo computacional da FCStyleGAN ao propagar *features* globais por toda a extensão da rede sintetizadora. Utilizando a mesma configuração de hardware (uma única GPU NVDIA V100 16GB), incluir o mecanismo de attention somente nas duas últimas camadas do gerador e discriminador da StyleGAN2 leva a uma alocação de memória maior do que o disponibilizado pela placa de vídeo. Dado isso, é necessário reduzir o *batch size* para somente 6 amostras, metade do valor suportado com a FCStyleGAN. Afim de atingir

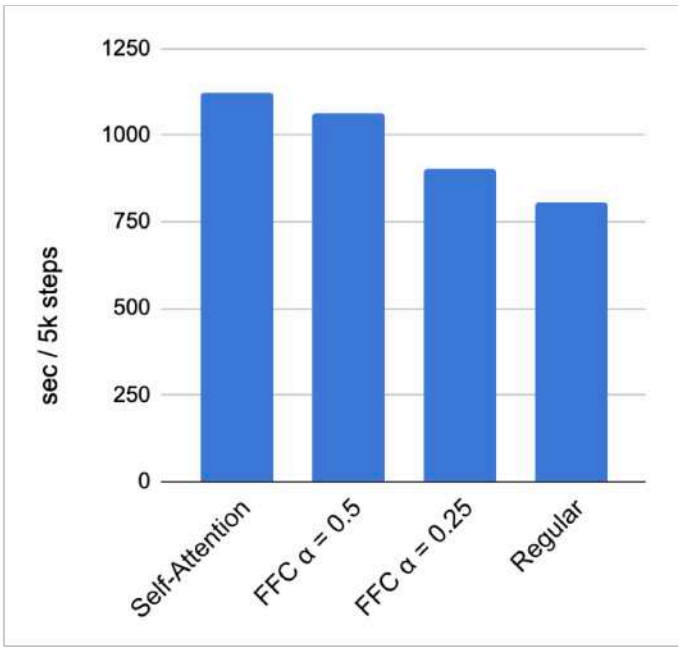

Figura 5.11: Comparação do custo computacional em segundos por 5 mil passos de treinamento. Resolução de 32×32. Experimento realizado em uma NVIDIA T4.

um número equivalente de atualizações do gradiente, a acumulação de gradiente foi alterada para 8 *mini-batches*. Nesse cenário, a iteração média com o mecanismo de attention custou 10.38 segundos por iteração. Enquanto a FCStyleGAN utilizando convoluções FFC demandou somente $7.66s/it$. Isto é, um tempo médio de iteração 26.2% menor.

5.8 Descrição dos experimentos na FCStyleGAN

Idealmente, operadores FFC devem ser adaptáveis a uma variedade de arquiteturas. Como discutido no Capítulo 3, trabalhos anteriores propuseram diversas arquiteturas para GANs. Entre elas, desta-case a família de StyleGANs, que alcançou um notável sucesso. Nesse experimento, utilizamos a rede FCStyleGAN descrita na Seção 4.3.1 com o objetivo de observar a atuação das features de Fourier em uma rede estado-da-arte. Devido ao alto custo computacional do treinamento da StyleGAN2, os experimentos tiveram um foco em resultados qualitativos. Modelos foram treinados para os conjuntos de dados CelebA e LSUN Bedroom na resolução 128×128, e FFHQ para a resolução 256×256.

5.8.1 Resultados Qualitativos da FCStyleGAN

Pode-se observar o resultado dos treinamentos em resolução 128×128 para os datasets LSUN Bedroom e CelebA na Figura 5.12 e na Figura 5.13, respectivamente. Em ambos os casos, os treinamentos foram realizados em uma única GPU NVIDIA V100. Para um *batch size* de 16, os modelos foram treinados por até 20 horas para alcançarem os resultados expostos. Assim como os resultados descritos na Seção 5.7, o modelo obteve melhores resultados com um $\alpha = 0.25$. Além disso, dado o maior número de parâmetros herdado pela StyleGAN2, o α menor permitiu um treinamento mais rápido.

Figura 5.12: Quartos geradas em 128×128 pela FCStyleGAN2 por meio do LSUN Bedroom.

Para as imagens de resolução 256×256, o treinamento foi realizado com todo o conjunto de dados FFHQ e em uma única GPU NVIDIA V100. Foi utilizado o mesmo valor de $\alpha = 0.25$, um *batch size* de 12, e a acumulação de gradiente por 4 atualizações. O treinamento ocorreu até 82k iterações. Os resultados desse experimento são apresentados na Figura 5.14. Como mencionado na Tabela 5.2, o FFHQ possui uma maior variedade de idade e etnicidade. Essa variedade também pode ser observada nas imagens geradas da Figura 5.14 e Figura 5.15, em comparação com os rotos do dataset CelebA

Figura 5.13: Rostos geradas em 128 × 128 pela FCStyleGAN2 com o dataset CelebA.

da Figura 5.13. Mesmo em uma resolução maior e em um conjunto de dados consideravelmente mais diverso, a FCStyleGAN foi capaz de gerar rostos realistas.

A análise de detalhes das imagens geradas pela FCStyleGAN revela a capacidade da rede em capturar e replicar nuances e características faciais distintas em uma resolução de 256 × 256. Detalhes como textura da pele, maquiagem, sombreamento, e expressões sutis são reproduzidos com fidelidade, aproximando as imagens geradas desejado realismo fotográfico. Na Figura 5.16, o rosto da primeira linha apresenta com consistência elementos de maquiagem, como batom, lápis de olho, além de sobrancelhas mais finas.

Por fim, também pode-se observar as aparentes dificuldades da FCStyleGAN ao gerar imagens realistas. Primeiro, em muitas das amostras das Figuras 5.14 e 5.15 observa-se fundos que apresentam padrões não realistas. Segundo, o modelo apresenta uma maior dificuldade ao representar cabelos longos, gerando padrões no fio que torna perceptível o fato de serem imagens geradas. Por último, as peças de roupas, quando aparentes, muitas vezes destoam da aparência real, misturando-se com as formas e cores do fundo da imagem.

5.8.2 Resultados Quantitativos da FCStyleGAN

Para a avaliação quantitativa da FCStyleGAN, foi priorizado o cálculo do FID para os modelos treinados a partir do dataset FFHQ. Dado o alto custo computacional do treinamento dos modelos e do cálculo do FID, métri-

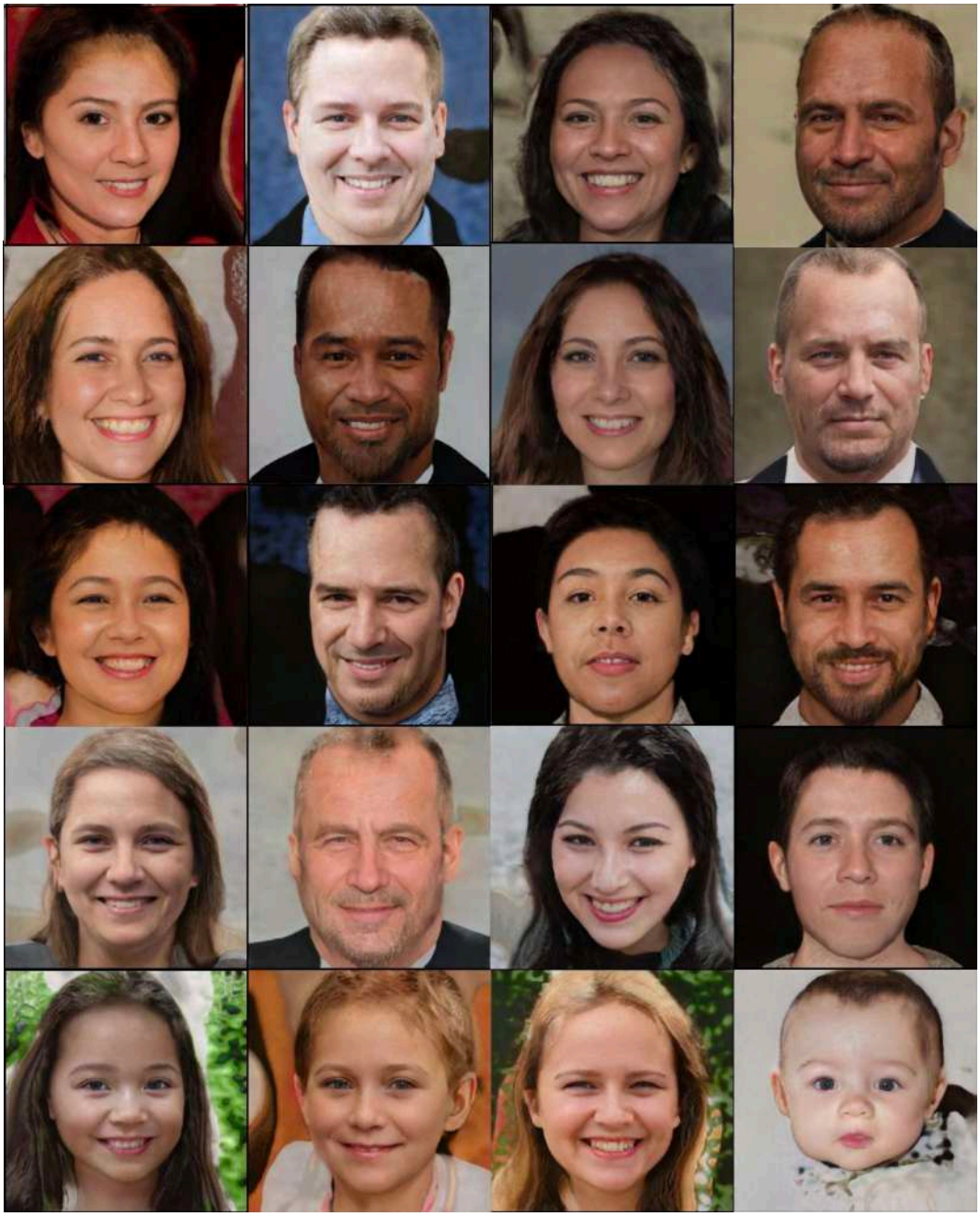

Figura 5.14: Rostos gerados em 256×256 pela FCStyleGAN2 a partir do conjunto de dados FFHQ.

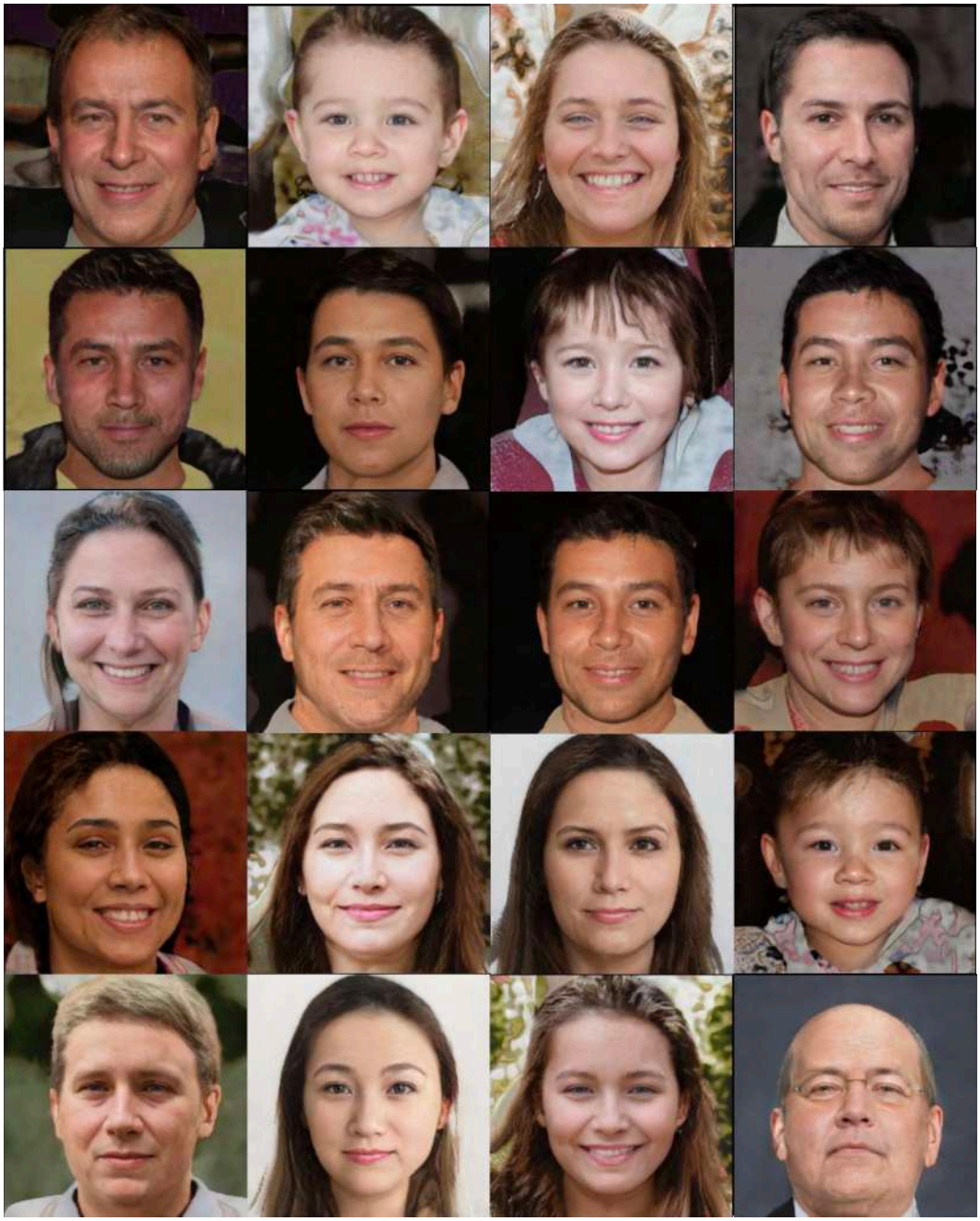

Figura 5.15: Rostos gerados em 256 × 256 pela FCStyleGAN2 a partir do conjunto de dados FFHQ.

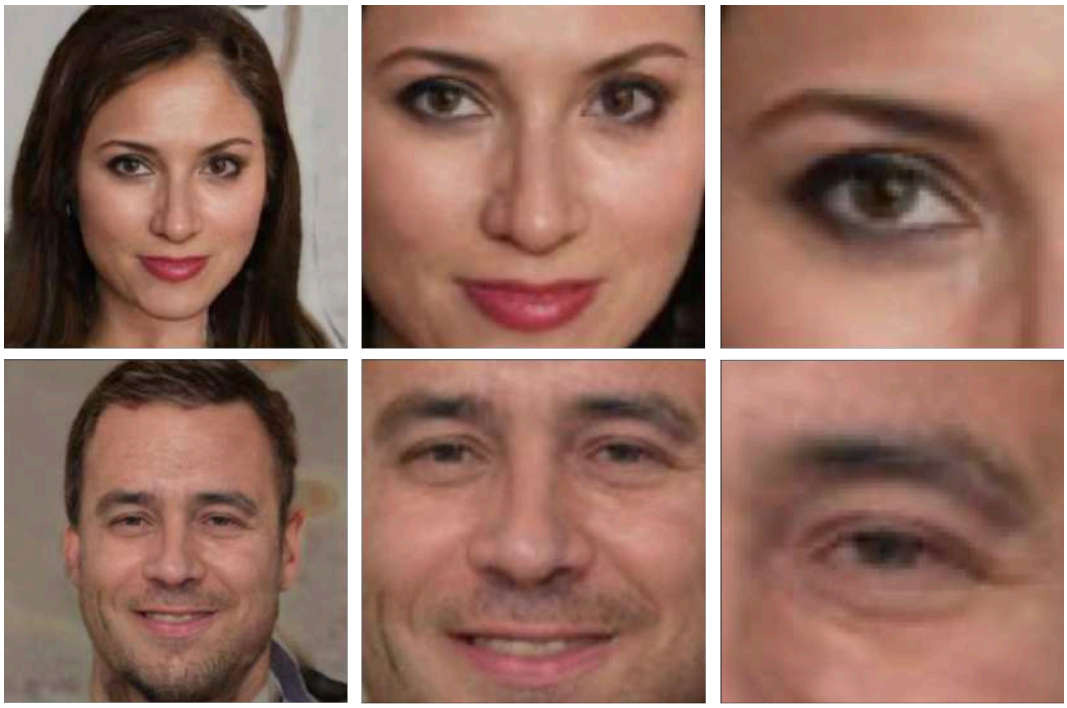

Figura 5.16: Detalhes de rostos gerados em resolução 256×256 pela FCStyleGAN2 a partir do conjunto de dados FFHQ.

cas quantitativas foram realizadas para o modelo que apresentou os melhores resultados qualitativos. O FID calculado levou em consideração 50 mil amostras dos dados reais e amostras geradas.

A Figura 5.17 mostra o comportamento da métrica de FID durante o treinamento do modelo FCStyleGAN. Esse treinamento foi realizado em uma única NVIDIA V100 16GB. O menor FID adquirido durante o treinamento foi 68.89, na iteração 82k. É possível observar pelo gráfico de treinamento que nesse momento o modelo ainda não atingiu o fim de seu treinamento, então é possível que em iterações futuras um valor menor de FID seja alcançado.

Uma explicação para o alto valor de FID é a qualidade não uniforme das imagens geradas pela rede. A Figura 5.18 apresenta alguns exemplos de amostras que são mais claramente apontadas como imagens falsas. Na primeira coluna da esquerda, imagens que apresentam rostos quase realistas, mas com defeitos evidentes na zona periférica. Na segunda coluna, rostos com aspecto realista, mas com elementos importantes, como o cabelo, não representados. A terceira e quarta coluna apresentam amostras que em que a rede não foi capaz de replicar os padrões presentes no conjunto de dados de treinamento.

Todas essas amostras foram adquiridas pelo modelo durante a mesma iteração. Isso mostra que ainda que a FCStyleGAN obtenha boa qualidade de geração para algumas amostras, ela não apresenta o mesmo comportamento

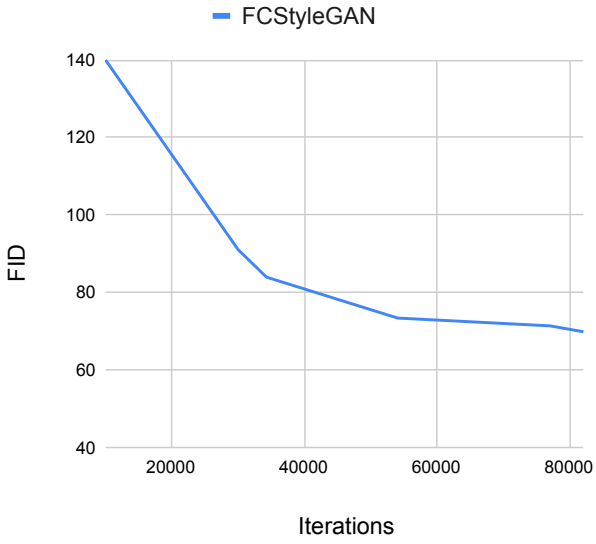

Figura 5.17: Amostras em 256 × 256 do conjunto de dados FFHQ.

para toda a distribuição dos dados reais. Além disso, o dataset FFHQ introduz o desafio de possuir imagens com peças de vestuário como bonés, capacetes, fantasias e véus. A FCStyleGAN aparenta não ter sido capaz de representar essa variedade em suas amostras, o que faz com que sejam geradas elementos amórficos, como mostra a Figura 5.19. À direita, amostras geradas pela FCStyleGAN que parecem tentar replicar acessórios para a cabeça. À esquerda, exemplos de imagens do conjunto de dados FFHQ que apresentam pessoas utilizando esses acessórios.

Figura 5.18: Exemplos de amostras não realistas geradas pela FCStyleGAN2 na iteração 82k.

Figura 5.19: Comparação entre imagens geradas pela FCStyleGAN (esquerda) e imagens do conjunto de dados FFHQ (direita).

5.8.3 Limitações experimentais

Dado o alto custo computacional atrelado, a configuração de rede como descrita na Seção 4.3.1 foi a única em que se realizou experimentos qualitativos e quantitativos. Trabalhos como a StyleGAN2 divulgam que o tempo de exploração inicial e de exploração para os resultados obtidos somam-se a mais que 66% do tempo de uso de GPU (KARRAS et al., 2020). Isso aponta para a necessidade de extensa exploração ao propor alterações em uma arquitetura do nível de complexidade da família StyleGAN.

Nos experimentos apresentados nessa Seção, não foram explorados métodos para ajuste de hiperparâmetros como *learning rate*, β_1 e β_2. Também não foram exploradas outras propostas para a introdução de operadores FFC. Assim, é possível que existam outras configurações que obtenham maior sucesso ao incluir convoluções espectrais na família das StyleGANs. Em especial, os experimentos para a proposta de bloco convolucional FFC adaptado à StyleGAN2 limitaram-se à arquitetura descrita na Figura 3.10.

A quantidade pequena de iterações em dimensionalidades maiores também pode ser considerado como uma limitação dos experimentos, dado que GANs notoriamente necessitam de um considerável tempo de treinamento. Dado o tempo limitado de acesso a GPUs, métricas quantitativas foram computadas somente para baixas dimensões ou modelos que obtiveram bom resultado qualitativo aparente.

6
Conclusão e trabalhos futuros

Nesse trabalho, exploramos a utilização do domínio da frequência para a tarefa de geração imagem por redes profundas. Apresentamos a Rede Generativa Adversarial com Convolução Rápida de Fourier (FCGAN), que utiliza operações de Fast Fourier Convolutions para incorporar features espectrais na geração de imagens em GANs. Em uma arquitetura de natureza instável, observamos que a estabilização do discriminador é suficiente para assegurar que o treinamento convirja mesmo com convoluções de escala global. Apresentamos evidências quantitativas e qualitativas para a afirmação que as features de Fourier auxiliam modelos geracionais a sintetizar imagens melhores. A FCGAN supera as redes originais que empregam apenas convoluções convencionais tanto no FID como no Inception Score nos testes com o CIFAR e STL-10.

Os resultados descritos foram alcançados em uma série de datasets benchmark de referência. Assim, argumentamos pela capacidade dos operadores de generalizarem para domínios distintos, seja para os com alta variância, seja para os que apresentam menos exemplos de amostra disponível. Mostramos também que os operadores de Fourier têm o potencial de capturar padrões estruturais. Esses operadores realizam essa tarefa com apenas um leve impacto no desempenho da rede, principalmente em comparação às tradicionais camadas de self-attention. Destaca-se, portanto, seu potencial como um operador global de baixo custo computacional. A utilização das convoluções espectrais em uma arquitetura como a StyleGAN2 também aponta para o potencial desses operadores para auxiliar modelos estado-da-arte.

A utilização de features de Fourier no discriminador levou à rápida discriminação de imagens reais e geradas. Trabalhos futuros podem explorar esse fenômeno com a utilização de Fast Fourier Convolutions para tarefas como ataques adversariais e classificação de imagens, campos que já encontraram sucesso ao incluir em seu treinamento o domínio espectral. A inclusão desses operadores em outras arquiteturas, como CycleGANs, que realizam a tarefa de image-to-image translation também pode ser explorada. Explorar a relação das operações no domínio da frequência para a tarefa de *transfer learning*, comum na família StyleGAN, é uma oportunidade para trabalhos futuros.

7
Referências bibliográficas

AN, G. The effects of adding noise during backpropagation training on a generalization performance. **Neural Computation**, v. 8, n. 3, p. 643–674, 1996.

ANANTRASIRICHAI, N.; BULL, D. Artificial intelligence in the creative industries: a review. **Artificial intelligence review**, Springer, p. 1–68, 2022.

ARJOVSKY, M.; CHINTALA, S.; BOTTOU, L. **Wasserstein GAN**. 2017.

AYERS, G.; DAINTY, J. C. Iterative blind deconvolution method and its applications. **Optics letters**, Optica Publishing Group, v. 13, n. 7, p. 547–549, 1988.

BARAHEEM, S. S.; LE, T.-N.; NGUYEN, T. V. Image synthesis: a review of methods, datasets, evaluation metrics, and future outlook. **Artificial Intelligence Review**, Springer, p. 1–53, 2023.

BENGIO, Y. et al. Deep generative stochastic networks trainable by backprop. In: PMLR. **International Conference on Machine Learning**. [S.l.], 2014. p. 226–234.

BIŃKOWSKI, M. et al. Demystifying mmd gans. **arXiv preprint arXiv:1801.01401**, 2018.

BROCK, A.; DONAHUE, J.; SIMONYAN, K. **Large Scale GAN Training for High Fidelity Natural Image Synthesis**. 2019.

CHANDRASEGARAN, K.; TRAN, N.-T.; CHEUNG, N.-M. A closer look at fourier spectrum discrepancies for cnn-generated images detection. In: **Proceedings of the IEEE/CVF conference on computer vision and pattern recognition**. [S.l.: s.n.], 2021. p. 7200–7209.

CHANDRASEGARAN, K.; TRAN, N.-T.; CHEUNG, N.-M. A closer look at fourier spectrum discrepancies for cnn-generated images detection. In: **Proceedings of the IEEE/CVF conference on computer vision and pattern recognition**. [S.l.: s.n.], 2021. p. 7200–7209.

CHI, L.; JIANG, B.; MU, Y. Fast fourier convolution. **Advances in Neural Information Processing Systems**, v. 33, p. 4479–4488, 2020.

COATES, A.; NG, A.; LEE, H. An analysis of single-layer networks in unsupervised feature learning. In: JMLR WORKSHOP AND CONFERENCE PROCEEDINGS. **Proceedings of the fourteenth international conference on artificial intelligence and statistics**. [S.l.], 2011. p. 215–223.

ELHARROUSS, O. et al. Image inpainting: A review. **Neural Processing Letters**, Springer, v. 51, p. 2007–2028, 2020.

FIENUP, J. R. Reconstruction of an object from the modulus of its fourier transform. **Optics letters**, Optica Publishing Group, v. 3, n. 1, p. 27–29, 1978.

FRANK, J. et al. Leveraging frequency analysis for deep fake image recognition. In: PMLR. **International conference on machine learning**. [S.l.], 2020. p. 3247–3258.

GLOROT, X.; BENGIO, Y. Understanding the difficulty of training deep feedforward neural networks. In: JMLR WORKSHOP AND CONFERENCE PROCEEDINGS. **Proceedings of the thirteenth international conference on artificial intelligence and statistics**. [S.l.], 2010. p. 249–256.

GOODFELLOW, I. et al. Generative adversarial nets. **NIPS**, v. 27, 2014.

GOODFELLOW, I. et al. Generative adversarial networks. **Communications of the ACM**, ACM New York, NY, USA, v. 63, n. 11, p. 139–144, 2020.

GUILLEMOT, C.; MEUR, O. L. Image inpainting: Overview and recent advances. **IEEE signal processing magazine**, IEEE, v. 31, n. 1, p. 127–144, 2013.

GULRAJANI, I. et al. Improved training of wasserstein gans. **Advances in neural information processing systems**, v. 30, 2017.

GUO, M.-H. et al. Attention mechanisms in computer vision: A survey. **Computational visual media**, Springer, v. 8, n. 3, p. 331–368, 2022.

HALL, E. L. et al. A survey of preprocessing and feature extraction techniques for radiographic images. **IEEE Transactions on Computers**, IEEE, v. 100, n. 9, p. 1032–1044, 1971.

HE, H. et al. Probgan: Towards probabilistic gan with theoretical guarantees. In: **International Conference on Learning Representations**. [S.l.: s.n.], 2019.

HENDRYCKS, D.; GIMPEL, K. **Gaussian Error Linear Units (GELUs)**. 2020.

HESSEL, J. et al. Clipscore: A reference-free evaluation metric for image captioning. **arXiv preprint arXiv:2104.08718**, 2021.

HEUSEL, M. et al. **GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium**. 2018.

HO, J.; JAIN, A.; ABBEEL, P. Denoising diffusion probabilistic models. **Advances in neural information processing systems**, v. 33, p. 6840–6851, 2020.

HU, J. et al. **Squeeze-and-Excitation Networks**. 2019.

HUANG, X.; BELONGIE, S. **Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization**. 2017.

IOFFE, S.; SZEGEDY, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: PMLR. **International conference on machine learning**. [S.l.], 2015. p. 448–456.

KARRAS, T. et al. **Progressive Growing of GANs for Improved Quality, Stability, and Variation**. 2018.

KARRAS, T. et al. **Training Generative Adversarial Networks with Limited Data**. 2020.

KARRAS, T. et al. Alias-free generative adversarial networks. In: **Proc. NeurIPS**. [S.l.: s.n.], 2021.

KARRAS, T.; LAINE, S.; AILA, T. **A Style-Based Generator Architecture for Generative Adversarial Networks**. 2019.

KARRAS, T. et al. **Analyzing and Improving the Image Quality of StyleGAN**. 2020.

KINGMA, D. P.; BA, J. **Adam: A Method for Stochastic Optimization**. 2017.

KINGMA, D. P.; WELLING, M. Auto-encoding variational bayes. **arXiv preprint arXiv:1312.6114**, 2013.

KRIZHEVSKY, A.; HINTON, G. et al. Learning multiple layers of features from tiny images. Toronto, ON, Canada, 2009.

KUMAR, A. et al. Chapter five - machine learning in medical imaging. In: FENG, D. D. (Ed.). **Biomedical Information Technology (Second Edition)**. Second edition. Academic Press, 2020, (Biomedical Engineering). p. 167–196. ISBN 978-0-12-816034-3. Disponível em: <https://www.sciencedirect.com/science/article/pii/B9780128160343000055>.

LARSEN, A. B. L. et al. Autoencoding beyond pixels using a learned similarity metric. In: PMLR. **International conference on machine learning**. [S.l.], 2016. p. 1558–1566.

LECUN, Y. et al. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, Ieee, v. 86, n. 11, p. 2278–2324, 1998.

LECUN, Y.; CORTES, C.; BURGES, C. Mnist handwritten digit database. **ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist**, v. 2, 2010.

LEE, K. et al. **ViTGAN: Training GANs with Vision Transformers**. 2021.

LEE, K. S.; TRAN, N.-T.; CHEUNG, N.-M. Infomax-gan: Improved adversarial image generation via information maximization and contrastive learning. In: **Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)**. [S.l.: s.n.], 2021. p. 3942–3952.

LEE-THORP, J. et al. Fnet: Mixing tokens with fourier transforms. **arXiv preprint arXiv:2105.03824**, 2021.

LI, B. et al. Controllable text-to-image generation. **Advances in Neural Information Processing Systems**, v. 32, 2019.

LI, J. et al. Frequency-aware discriminative feature learning supervised by single-center loss for face forgery detection. In: **Proceedings of the IEEE/CVF conference on computer vision and pattern recognition**. [S.l.: s.n.], 2021. p. 6458–6467.

LIM, J. H.; YE, J. C. **Geometric GAN**. 2017.

LITJENS, G. et al. A survey on deep learning in medical image analysis. **Medical image analysis**, Elsevier, v. 42, p. 60–88, 2017.

LIU, Z. et al. Deep learning face attributes in the wild. In: **Proceedings of the IEEE international conference on computer vision**. [S.l.: s.n.], 2015. p. 3730–3738.

LIU, Z. et al. **A ConvNet for the 2020s**. 2022.

LONG, W. et al. Image colorization with fast fourier convolution. In: **Proceedings of the 2023 7th International Conference on Innovation in Artificial Intelligence**. [S.l.: s.n.], 2023. p. 60–65.

LOSHCHILOV, I.; HUTTER, F. **Decoupled Weight Decay Regularization**. 2019.

LU, Z. et al. Glama: Joint spatial and frequency loss for general image inpainting. In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops**. [S.l.: s.n.], 2022. p. 1301–1310.

LUO, W. et al. Understanding the effective receptive field in deep convolutional neural networks. **Advances in neural information processing systems**, v. 29, 2016.

MATHIEU, M.; HENAFF, M.; LECUN, Y. Fast training of convolutional networks through ffts. **arXiv preprint arXiv:1312.5851**, 2013.

MILDENHALL, B. et al. Nerf: Representing scenes as neural radiance fields for view synthesis. **Communications of the ACM**, ACM New York, NY, USA, v. 65, n. 1, p. 99–106, 2021.

MIRZA, M.; OSINDERO, S. Conditional generative adversarial nets. **arXiv preprint arXiv:1411.1784**, 2014.

MIYATO, T. et al. Spectral normalization for generative adversarial networks. **arXiv preprint arXiv:1802.05957**, 2018.

NETZER, Y. et al. Reading digits in natural images with unsupervised feature learning. 2011.

NGUYEN, T. et al. Dual discriminator generative adversarial nets. **Advances in neural information processing systems**, v. 30, 2017.

NILSBACK, M.-E.; ZISSERMAN, A. Automated flower classification over a large number of classes. In: **Indian Conference on Computer Vision, Graphics and Image Processing**. [S.l.: s.n.], 2008.

OBUKHOV, A. et al. **High-fidelity performance metrics for generative models in PyTorch**. Zenodo, 2020. Version: 0.3.0, DOI: 10.5281/zenodo.4957738. Disponível em: <https://github.com/toshas/torch-fidelity>.

ODENA, A.; OLAH, C.; SHLENS, J. Conditional image synthesis with auxiliary classifier gans. In: PMLR. **International conference on machine learning**. [S.l.], 2017. p. 2642–2651.

PRATT, H. et al. Fcnn: Fourier convolutional neural networks. In: SPRINGER. **Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2017, Skopje, Macedonia, September 18–22, 2017, Proceedings, Part I 17**. [S.l.], 2017. p. 786–798.

QIAO, T. et al. **MirrorGAN: Learning Text-to-image Generation by Redescription**. 2019.

RADFORD, A. et al. Learning transferable visual models from natural language supervision. In: PMLR. **International conference on machine learning**. [S.l.], 2021. p. 8748–8763.

RADFORD, A.; METZ, L.; CHINTALA, S. **Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks**. 2016.

RAMACHANDRAN, P. et al. Stand-alone self-attention in vision models. **Advances in neural information processing systems**, v. 32, 2019.

RAMESH, A. et al. Zero-shot text-to-image generation. In: PMLR. **International Conference on Machine Learning**. [S.l.], 2021. p. 8821–8831.

RAO, Y. et al. Global filter networks for image classification. **Advances in neural information processing systems**, v. 34, p. 980–993, 2021.

REED, S. et al. Generative adversarial text to image synthesis. In: PMLR. **International conference on machine learning**. [S.l.], 2016. p. 1060–1069.

RIPPEL, O.; SNOEK, J.; ADAMS, R. P. Spectral representations for convolutional neural networks. **Advances in neural information processing systems**, v. 28, 2015.

ROMBACH, R. et al. High-resolution image synthesis with latent diffusion models. In: **Proceedings of the IEEE/CVF conference on computer vision and pattern recognition**. [S.l.: s.n.], 2022. p. 10684–10695.

SAJJADI, M. S. et al. Assessing generative models via precision and recall. **Advances in neural information processing systems**, v. 31, 2018.

SALIMANS, T. et al. **Improved Techniques for Training GANs**. 2016.

SAUER, A. et al. Stylegan-t: Unlocking the power of gans for fast large-scale text-to-image synthesis. **arXiv preprint arXiv:2301.09515**, 2023.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. **arXiv preprint arXiv:1409.1556**, 2014.

SINHA, A. K.; MOORTHI, S. M.; DHAR, D. Nl-ffc: Non-local fast fourier convolution for image super resolution. In: **2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)**. [S.l.: s.n.], 2022. p. 466–475.

SKOROKHODOV, I.; IGNATYEV, S.; ELHOSEINY, M. **Adversarial Generation of Continuous Images**. 2021.

SØNDERBY, C. K. et al. Amortised map inference for image super-resolution. **arXiv preprint arXiv:1610.04490**, 2016.

SUVOROV, R. et al. **Resolution-robust Large Mask Inpainting with Fourier Convolutions**. 2021.

TANCIK, M. et al. Fourier features let networks learn high frequency functions in low dimensional domains. **Advances in Neural Information Processing Systems**, v. 33, p. 7537–7547, 2020.

THEIS, L.; OORD, A. v. d.; BETHGE, M. A note on the evaluation of generative models. **arXiv preprint arXiv:1511.01844**, 2015.

VASWANI, A. et al. **Attention Is All You Need**. 2017.

VOULODIMOS, A. et al. Deep learning for computer vision: A brief review. **Computational intelligence and neuroscience**, Hindawi, v. 2018, 2018.

VRIES, H. de et al. **Modulating early visual processing by language**. 2017.

WANG, X. et al. Non-local neural networks. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2018. p. 7794–7803.

YU, F. et al. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. **arXiv preprint arXiv:1506.03365**, 2015.

ZHANG, H. et al. Self-attention generative adversarial networks. **CoRR**, abs/1805.08318, 2018.