UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO

MAURÍCIO BARBOSA DA ROCHA

Aceleração do Processo de Edição de Imagens Baseadas em StyleGANs

Monografia apresentada como requisito parcial para a obtenção do grau de Bacharel em Engenharia de Computação

Orientador: Prof. Dr. Manuel Menezes de Oliveira Neto

Porto Alegre
2024

AGRADECIMENTOS

Ao encerrar esta etapa fundamental da minha vida acadêmica, não posso deixar de expressar minha profunda gratidão àqueles que contribuíram para a realização deste trabalho.

Primeiramente, à Universidade Federal do Rio Grande do Sul, pelo ambiente acadêmico enriquecedor e pelo corpo docente excepcional que me proporcionou conhecimento e inspiração durante toda a minha jornada educacional.

Um agradecimento especial ao meu orientador, Prof. Manuel Oliveira, cuja sabedoria, paciência e rigor acadêmico foram decisivos para o desenvolvimento e aprimoramento deste trabalho. Sua orientação foi mais do que acadêmica; foi uma lição de dedicação e paixão pela pesquisa.

À minha família, aos meus queridos pais, Sandra e José Maurício, e à minha irmã Savanah, pela inabalável fé em minhas capacidades e pelo apoio incondicional em cada passo que dei, tanto na vida acadêmica quanto fora dela. Vocês foram meu porto seguro, minha fonte de encorajamento e amor.

À minha esposa, Vitória, pelo companheirismo, compreensão e por todas as vezes que me incentivou a seguir em frente. Sua presença foi essencial para que eu pudesse superar os desafios e alcançar este momento.

Este trabalho é também um reflexo do apoio e do carinho que recebi de cada um de vocês. Meu sincero obrigado.

RESUMO

Este trabalho explora o potencial dos modelos generativos, em particular as Generative Adversarial Networks (GANs), no contexto da edição de imagens. Ao manipular vetores de características, essas técnicas possibilitam a criação de imagens fotorrealistas e oferecem uma abordagem mais intuitiva e controlada para a edição de atributos específicos. Por meio de experimentos e análises, este estudo concentra-se em aprimorar a técnica DragGAN, uma abordagem de edição de imagens baseada na arquitetura StyleGAN2. Nosso objetivo é melhorar a eficiência na manipulação de vetores de características, especialmente na etapa de Supervisão de Movimentos da DragGAN. Desenvolvemos um método alternativo para a otimização de vetores de características, que analisa a diferença entre os vetores de características iniciais e os obtidos após as três primeiras iterações utilizando o método tradicional. Após isso, o novo método é aplicado nas iterações subsequentes até que a distância entre os pontos de manipulação e alvo deixe de apresentar reduções. Quando isso ocorre, ajustes finos são realizados utilizando o método tradicional. Adicionalmente, propomos alterações no *learning rate* e no raio do Rastreamento de Pontos. Comparada à abordagem tradicional, nossa solução acelera o processo de manipulação interativa de imagens através da técnica DragGAN.

Palavras-chave: StyleGAN2. DragGAN. modelos generativos. espaço latente. manipulação do espaço latente.

ABSTRACT

This work explores the potential of generative models, particularly Generative Adversarial Networks (GANs), in the context of image editing. By manipulating feature vectors, these techniques enable the creation of photorealistic images and offer a more intuitive and controlled approach to editing specific attributes. Through experiments and analysis, this study focuses on enhancing the DragGAN technique, an image editing approach based on the StyleGAN2 architecture. Our goal is to improve the efficiency in manipulating latent vectors, especially in the Motion Supervision stage of DragGAN. We have developed an alternative method for optimizing latent vectors, which analyzes the difference between the initial latent vectors and those obtained after the first three iterations using the traditional method. Thereafter, the new method is applied in subsequent iterations until the distance between the manipulation points and the target ceases to show reductions. When this occurs, fine adjustments are made using the traditional method. Additionally, we propose changes to the learning rate and the radius of Point Tracking. Compared to the traditional approach, our solution accelerates the process of interactive image manipulation through the DragGAN technique.

Keywords: StyleGAN2, DragGAN, Image Generation, Latent Space, Latent Manipulation.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

As técnicas de edição de imagens fundamentadas em modelos generativos têm se destacado como uma abordagem inovadora e promissora. Mediante a aplicação de modelos generativos baseados nas *Generative Adversarial Networks* (GANs) (GOODFELLOW et al., 2014), em particular a StyleGAN2 (KARRAS et al., 2020), torna-se possível criar e editar imagens fotorrealistas através da manipulação de vetores de características. Esta capacidade abre novas perspectivas na edição e criação de conteúdo visual de modo mais intuitivo e controlável, com aplicabilidades que vão desde design gráfico até a produção de conteúdo para jogos e cinema.

Além da StyleGAN2, uma técnica específica que ganhou atenção é a DragGAN (PAN et al., 2023), que se baseia na arquitetura da StyleGAN2 e permite uma manipulação ainda mais refinada de imagens. Este trabalho visa acelerar a técnica DragGAN, abordando desafios específicos em sua eficiência e controle no processo de edição. Para tal, propomos modificações na metodologia original.

Focando especialmente na etapa de Supervisão de Movimentos da DragGAN, implementamos as principais modificações. Desenvolvemos um método alternativo para a otimização de vetores de características, baseando-nos na análise da direção inicial do movimento no espaço latente. Isto é obtido a partir das diferenças entre os vetores de características iniciais e aqueles após as três primeiras otimizações com o método tradicional. Inicialmente, realizam-se três iterações com o método tradicional, seguidas pela aplicação do método proposto até que a distância entre os pontos de manipulação e alvo deixe de diminuir. Neste ponto, ajustes finos poderão ser efetuados utilizando o método tradicional novamente. Adicionalmente, foram realizados ajustes à taxa de aprendizagem (*learning rate*) e no raio utilizado na etapa de rastreamento dos pontos, visando uma maior precisão e eficácia na edição.

Essas modificações tornaram possível acelerar a manipulação interativa de imagens baseadas em modelos generativos. A abordagem proposta demonstrou ser aproximadamente 22% mais rápida do que a abordagem tradicional para manipulação interativa de imagens por meio da técnica DragGAN, quando se utiliza um par de pontos (origem e destino) para especificar a edição. Além desse avanço, a realização deste trabalho permitiu a familiarização e o entendimento de um conjunto de técnicas do estado da arte ligadas ao uso de modelos generativos para síntese e manipulação de imagens fotorealísticas.

O restante deste trabalho encontra-se organizado da seguinte forma: o Capítulo 2

descreve as técnicas StyleGAN2, UserControllableLT e DragGAN, provendo a fundamentação teórica para a compreensão do trabalho desenvolvido. O Capítulo 3 apresenta a solução proposta para aceleração da técnica interativa de edição de imagens DragGAN. O Capítulo 4 discute os resultados obtidos com a implementação da técnica proposta, enquanto o Capítulo 5 apresenta algumas conclusões e possibilidades para trabalhos futuros.

2 REVISÃO BIBLIOGRÁFICA

As técnicas exploradas na revisão bibliográfica têm como base a arquitetura StyleGAN2 (KARRAS et al., 2020). Para um melhor entendimento do tema em discussão, é relevante fornecer uma descrição concisa dessa estrutura. Essa descrição será apresentada na Seção 2.1. Além disso, nesta revisão, abordaremos a técnica UserControllableLT (ENDO, 2022), precursora da técnica DragGAN (PAN et al., 2023) e utilizada como linha de base para o seu desenvolvimento. Posteriormente, na Seção 2.3, introduziremos a técnica DragGAN, uma abordagem interativa de edição de imagens baseada em modelos generativos. A Seção 2.4 incluirá um comparativo entre as diferentes técnicas analisadas. Por fim, na Seção 2.5, serão discutidas limitações da técnica DragGAN, algumas das quais este estudo visa superar.

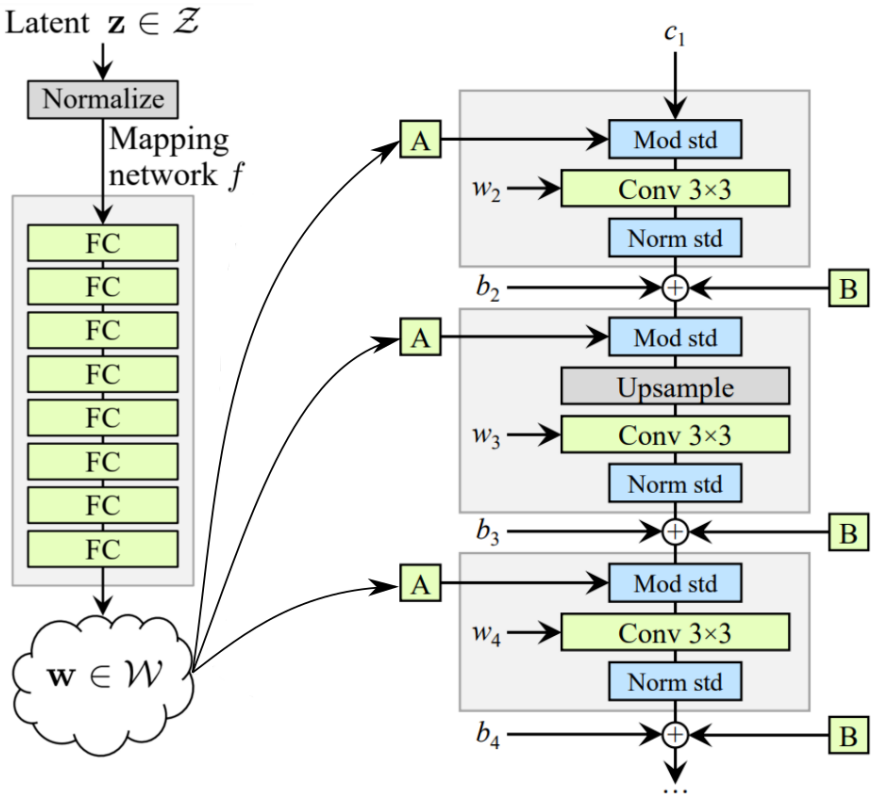
2.1 StyleGAN2

A StyleGAN2 é uma abordagem fundamentada nas GANs (GOODFELLOW et al., 2014). GANs são arquiteturas compostas por uma rede geradora G, responsável por aprender a distribuição dos dados de treinamento para gerar novas amostras, e uma rede discriminadora D, que estima a probabilidade de uma amostra pertencer aos dados sintéticos ou aos dados reais. Essas redes operam em um processo adversarial, no qual a rede geradora busca produzir amostras cada vez mais realistas, enquanto a rede discriminadora busca melhorar sua capacidade de distinguir entre amostras reais e sintéticas. Essa competição entre as duas redes leva ao aprimoramento da qualidade das amostras geradas ao longo do treinamento. Uma GAN pode ser aplicada na síntese de imagens, convertendo vetores de características amostrados aleatoriamente em imagens fotorrealistas.

O objetivo da StyleGAN2 é gerar imagens fotorrealistas, permitindo aos usuários controlar diversos atributos nas imagens geradas, como estilo e aparência. A Figura 2.1 demonstra a arquitetura da rede de mapeamento e da rede geradora da StyleGAN2. No modelo proposto, um vetor latente z de 512 dimensões é obtido a partir de uma distribuição normal multivariada $\mathcal{N}$, com média zero e uma matriz de covariância equivalente à identidade. Esse vetor latente z é então mapeado para um vetor intermediário $w \in \mathbb{R}^{512}$ através de uma rede de mapeamento f. Esse mapeamento tem como objetivo aprender uma representação latente que seja facilmente controlável, permitindo a manipulação de características específicas na imagem gerada. O espaço de vetores w é denotado por $\mathcal{W}$.

O vetor latente w passa por transformações afins e é combinado com variações estocásticas obtidas pela adição de mapas de ruído aleatório. Essa combinação é alimentada na rede geradora G para produzir a imagem $I = G(w)$. A rede geradora é uma arquitetura em cascata que permite controlar diferentes escalas de detalhes nas imagens geradas. O vetor latente w é copiado e enviado para diferentes camadas da rede geradora G para controlar diferentes níveis de atributos, contribuindo com detalhes importantes para o fotorrealismo da imagem gerada. A Figura 2.2 apresenta alguns exemplos de imagens geradas pela StyleGAN2 utilizando o conjunto de dados FFHQ (Flickr-Faces-HQ), que consiste em fotografias de alta qualidade de rostos humanos.

Figura 2.1: Arquitetura da rede de mapeamento e da rede geradora da StyleGAN2: a rede de mapeamento consiste em oito camadas totalmente conectadas (*fully-connected*), enquanto a rede geradora é composta por dezoito camadas, distribuídas com duas camadas para cada nível de resolução, partindo de 4^2 até 1024^2. Na figura, 'A' representa as transformações afins aplicadas a cada nível e 'B' indica o ruído adicionado em cada camada da rede geradora.



Fonte: adaptada pelo autor de (KARRAS; LAINE; AILA, 2019) e (KARRAS et al., 2020)

Figura 2.2: Exemplos de imagens geradas com a StyleGAN2 utilizando o conjunto de dados FFHQ (Flickr-Faces-HQ).

Fonte: (KARRAS et al., 2020)

2.2 User-Controllable Latent Transformer for StyleGAN Image Layout Editing

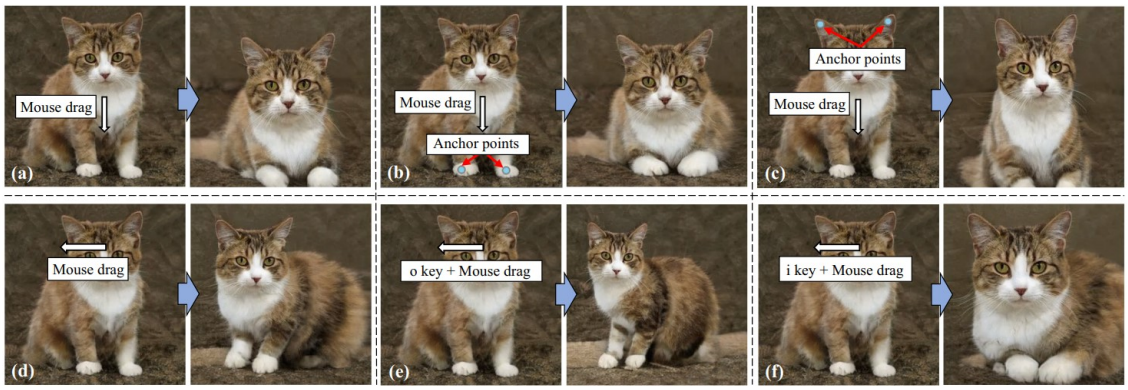
O método UserControllableLT (ENDO, 2022) propõe uma abordagem para viabilizar a manipulação interativa de vetores de características em uma arquitetura StyleGAN2, levando em conta as entradas fornecidas pelo usuário. Nessa técnica, o usuário pode selecionar quais pontos da imagem deseja manter inalterados e quais deseja tornar ajustáveis, além de definir a direção do movimento ao arrastar o mouse.

Na Figura 2.3, é possível observar exemplos de manipulações interativas realizadas pelo usuário. Conforme ilustrado em (a), o usuário define um vetor de movimento no gato da imagem ao mover o mouse, buscando identificar uma direção latente para orientar o movimento. O sistema exibe em tempo real os resultados da edição, movendo os vetores de características nas direções encontradas a partir do movimento do mouse. Em (b) e (c), foi exemplificada a inserção de pontos de ancoragem, que representam pontos que

se deseja manter imóveis durante o processo de edição.

Além da possibilidade de movimento em 2D demonstrado em (d), o sistema possibilita a simulação de movimento tridimensional (3D) por meio da combinação do movimento do mouse com o pressionar das teclas ‘i’, para realizar *zoom in*, ou ‘o’, para *zoom out* na imagem, conforme ilustrado em (e) e (f).

Figura 2.3: Edição do layout de imagens utilizando o transformador latente controlável pelo usuário.



Fonte: (ENDO, 2022)

As informações fornecidas pelo usuário e os vetores de características iniciais são encaminhados para um transformador latente, fundamentado em uma arquitetura codificador-decodificador baseada em transformadores (*transformers*). Esse transformador latente estima os vetores de características de saída, que são então utilizados pela rede geradora da StyleGAN2 para criar uma imagem resultante. Desse modo, o método possibilita ao usuário um controle mais preciso sobre a manipulação da imagem gerada pela StyleGAN2, permitindo personalizar sua aparência de acordo com suas preferências.

O transformador latente T transforma os vetores de características iniciais w_{antes} de acordo com as entradas de usuário $\mathcal{U}$ conforme a equação:

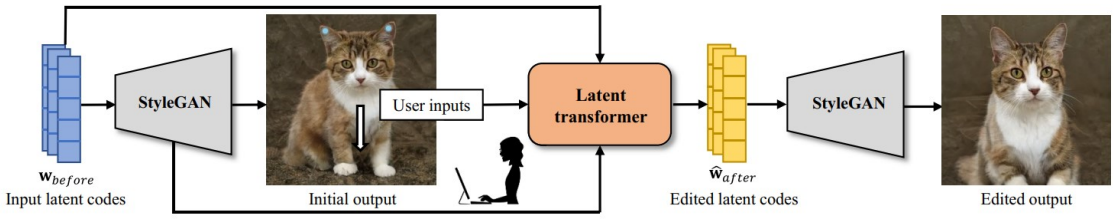
$$T(w_{antes}, \mathcal{U}, \alpha) = w_{antes} + \alpha \cdot f(w_{antes}, \mathcal{U}), \tag{2.1}$$

onde α é um parâmetro que ajusta o grau de manipulação do vetor latente w_{antes}, e f é uma função complexa implementada por uma rede neural que emprega componentes do modelo transformador, como codificação e decodificação, para processar e integrar as entradas de usuário $\mathcal{U}$ e os vetores latentes. As entradas de usuário são definidas como $\mathcal{U} = \{v_i, p_i\}_{i=1}^{K}$, onde K representa a quantidade de vetores de movimento $v_i \in \mathbb{R}^3$. As duas primeiras dimensões de v_i originam-se do vetor no plano 2D da imagem, que indica onde o usuário inseriu o vetor de movimento. A terceira dimensão é adquirida através

do pressionamento das teclas ‘i’ para *zoom in* e ‘o’ para *zoom out*. Por sua vez, $p_i \in \mathbb{Z}^2$ denota as posições dos *pixel* nos pontos iniciais de v_i.

A Figura 2.4 ilustra o fluxo de edição interativa das imagens usando o transformador latente. Inicialmente, o usuário especifica os vetores de movimento diretamente na imagem gerada pela StyleGAN2, indicando a orientação do movimento desejado na imagem. O transformador latente, então, calcula os vetores de características de saída, usando como entrada os vetores de características, as informações fornecidas pelo usuário e os mapas de características da StyleGAN2. Como resultado são obtidos os vetores de características $\hat{w}_{depois}$. Esses vetores de características resultantes são, por fim, utilizados no gerador da StyleGAN2 para gerar a imagem editada.

Figura 2.4: Processo de edição de imagem utilizando o método UserControllableLT. Vetores de características w_before de entrada serão processados pela StyleGAN2 para produzir a saída inicial. O usuário então fornece entradas que são aplicadas ao transformador latente juntamente com os mapas de características da StyleGAN2, resultando em vetores de características w_after editados. Estes vetores são novamente processados pela StyleGAN2 para gerar a saída editada.



Fonte: (ENDO, 2022)

Com o intuito de acomodar diferentes quantidades de entradas do usuário, foi empregada uma abordagem codificador-decodificador baseada na arquitetura dos transformadores. Essa estrutura tem a capacidade de lidar com entradas de extensões variáveis no contexto do transformador latente.

No âmbito do codificador do transformador latente, a partir das entradas $\mathcal{U}$ do usuário, uma sequência de vetores de características é extraída dos mapas de características da StyleGAN2. Após essa etapa, as sequências de vetores de movimento e de características são combinadas e transmitidas ao codificador do transformador. Nesse ponto, a camada de auto-atenção (*self-attention*) é empregada no codificador do transformador, usando a sequência resultante do passo anterior para extrair características globais. Isso possibilita a captura das relações entre múltiplas entradas de usuário.

Na perspectiva do decodificador do transformador, a saída do codificador e o vetor latente w_{antes} são utilizados para calcular o vetor latente $\hat{w}_{depois}$. Nesta abordagem,

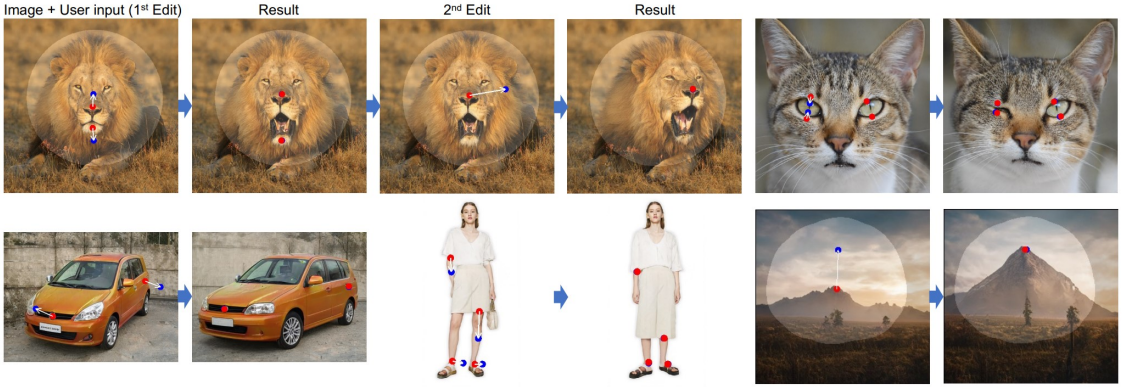
assume-se que é viável utilizar um vetor latente distinto para controlar cada camada da StyleGAN2. Deste modo, o decodificador do transformador recebe uma sequência de vetores de características como entrada. Esta sequência é combinada com vetores de incorporação de posição, que representam informações sobre a posição de cada elemento na sequência de entrada. Os vetores de incorporação de posição são parâmetros passíveis de aprendizado e auxiliam na diferenciação dos vetores de características para cada camada. Posteriormente, o decodificador do transformador extrai características que capturam as interações entre as entradas do usuário e os vetores de características. As direções de características são então calculadas a partir da saída do decodificador do transformador por meio de uma camada linear. Por fim, o vetor latente $\hat{w}_{depois}$ é obtido escalando as direções de características pelo fator α e somando-as ao vetor latente inicial w_{antes}. Esse vetor latente resultante é então empregado na StyleGAN2 para gerar a imagem com as modificações desejadas.

2.3 Drag Your GAN: Interactive Point-based Manipulation on the Generative Image Manifold

DragGAN (PAN et al., 2023) é uma abordagem interativa de edição de imagens baseada em GANs. Essa abordagem possibilita aos usuários selecionar um conjunto de pontos de manipulação e pontos alvo e, em seguida, mover precisamente cada ponto de manipulação para o seu ponto alvo correspondente. Conforme mostrado na Figura 2.5, essa manipulação controlada dos pontos permite a edição de diversos atributos espaciais, tais como pose, forma, expressão e *layout*, abrangendo uma ampla gama de categorias de objetos. A abordagem oferece a possibilidade de criar uma máscara na imagem editada, o que permite delimitar regiões passíveis de edição.

A DragGAN é composta por dois componentes principais. O primeiro componente consiste em uma *Supervisão de Movimentos* baseada em vetores de características (*features*), que orienta o movimento dos pontos de manipulação em direção aos pontos alvo correspondentes. O segundo componente consiste em uma abordagem de *Rastreamento de Pontos*, cujo propósito é facilitar a localização precisa dos pontos de manipulação e acompanhar sua posição durante a edição. A técnica utilizada na DragGAN baseia-se no fato de que o espaço de características de uma GAN contém informações suficientes para permitir tanto a Supervisão de Movimentos quanto o Rastreamento de Pontos.

Figura 2.5: Manipulação controlada de pontos. Pontos indicados em vermelho representam pontos de manipulação, enquanto os pontos em azul correspondem a pontos alvo. Os pontos de manipulação são deslocados em direção aos pontos alvo, modificando os vetores de características e ocasionando a edição das imagens correspondentes.

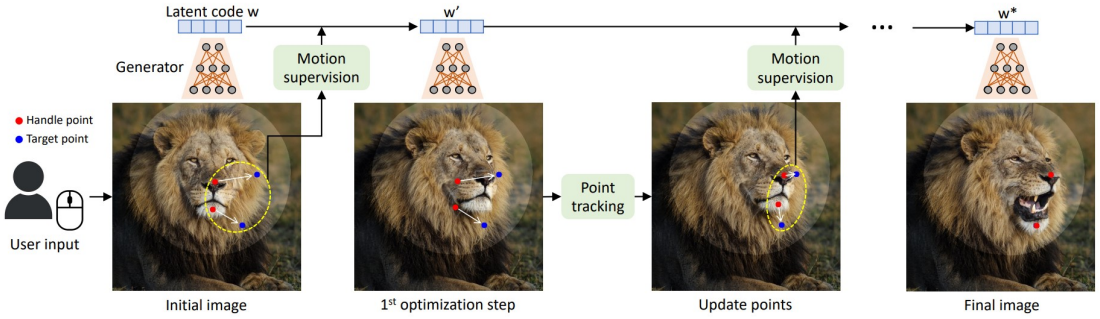


Fonte: (PAN et al., 2023)

Uma visão geral do *pipeline* de manipulação pode ser observada na Figura 2.6. Para qualquer imagem $I \in \mathbb{R}^{3\times H\times W}$ gerada por uma GAN, que possua um vetor latente w, o usuário tem a opção de selecionar um conjunto de pontos de manipulação $\{p_i = (x_{p,i}, y_{p,i}) | i = 1, 2, ..., n\}$, juntamente com seus pontos alvo correspondentes $\{t_i = (x_{t,i}, y_{t,i}) | i = 1, 2, ..., n\}$. O usuário pode, opcionalmente, desenhar uma máscara binária M, indicando as regiões da imagem passíveis de movimentação. O objetivo é mover o objeto na imagem de forma que suas posições semânticas (como o nariz e a mandíbula, ilustrados na Figura 2.6) atinjam seus pontos alvo correspondentes. Após a especificação dos pares de pontos de manipulação e alvo fornecidos pelo usuário, serão realizadas manipulações na imagem de maneira otimizada. Conforme ilustrado na Figura 2.6, cada etapa de otimização é composta sequencialmente pela Supervisão de Movimentos, seguida pelo Rastreamento de Pontos.

A etapa de Supervisão de Movimentos direciona cada ponto de manipulação para realizar um pequeno deslocamento em direção ao seu ponto alvo correspondente. Conforme ilustrado na Figura 2.7, para realizar o movimento de um ponto de manipulação p_i em direção a um ponto alvo t_i, é realizada a Supervisão de Movimentos de um pequeno *patch* em torno de p_i (representado pelo círculo vermelho), fazendo com que ele se mova em direção a t_i (representado pelo círculo azul) em um pequeno passo. Foi utilizado $\Omega_1(p_i, r_1)$ para representar os *pixels* cuja distância para p_i é menor do que r_1. A função

Figura 2.6: Visão geral do *pipeline* de manipulação da DragGAN. A partir de um vetor de características w, gera-se a imagem inicial. O usuário então interage com esta imagem, inserindo pares de pontos de manipulação (pontos em vermelho) e pontos alvo (pontos em azul). Após essa interação, inicia-se a etapa de Supervisão de Movimentos, que produz um novo vetor de características w', resultando em uma nova imagem com as modificações desejadas. A etapa de Rastreamento de Pontos se segue, atualizando as posições dos pontos de manipulação de acordo com as novas posições na imagem gerada. Esse processo continua até que cada ponto de manipulação atinja seu respectivo ponto alvo.



Fonte: (PAN et al., 2023)

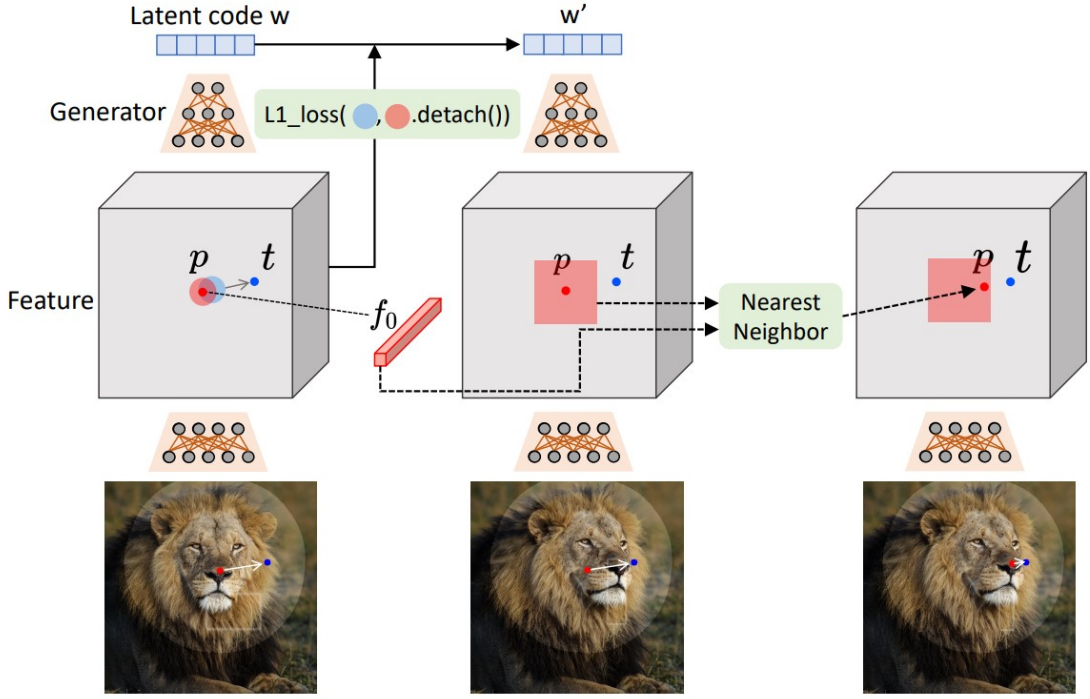
de perda criada para a Supervisão de Movimentos é calculada como:

$$\mathcal{L} = \sum_{i=0}^{n} \sum_{\mathbf{q_j} \in \Omega_1(\mathbf{p_i}, r_1)} \|\mathbf{F}(\mathbf{q_j}) - \mathbf{F}(\mathbf{q_j} + \mathbf{d_i})\|_1 + \lambda \|(\mathbf{F} - \mathbf{F_0}) \cdot (1 - M)\|_1, \tag{2.2}$$

onde $\mathbf{F}(\mathbf{q})$ representa os valores das características de $\mathbf{F}$ no *pixel* $\mathbf{q}$ em todas as camadas do sexto bloco da StyleGAN2, $\mathbf{d_i} = \frac{\mathbf{t_i} - \mathbf{p_i}}{\|\mathbf{t_i} - \mathbf{p_i}\|_2}$ é um vetor normalizado que aponta de p_i para t_i ($\mathbf{d_i} = 0$ se $t_i = p_i$), $\mathbf{F_0}$ é o mapa de características de todas as camadas do sexto bloco da StyleGAN2 correspondentes à imagem inicial e n é a quantidade total de pares de pontos informados pelo usuário. Como os componentes $\mathbf{q_j} + \mathbf{d_i}$ não são inteiros, $\mathbf{F}(\mathbf{q_j} + \mathbf{d_i})$ é obtido via interpolação bilinear. A cada passo da Supervisão de Movimentos, o valor de $\mathcal{L}$ é utilizado para otimizar o vetor latente w por meio do otimizador Adam (KINGMA; BA, 2014).

A Supervisão de Movimentos mencionada anteriormente resulta na obtenção de um novo vetor latente w', um novo mapa de características F' e uma nova imagem I'. Essa atualização movimenta cada ponto de manipulação em direção ao seu alvo em um pequeno passo. O comprimento exato desse passo de otimização está sujeito a uma dinâmica de otimização complexa e pode variar para diferentes categorias de objetos. Na etapa subsequente, é realizado o Rastreamento de Pontos, responsável por atualizar as posições de cada ponto de manipulação $\{p_i\}$ para que eles rastreiem corretamente os pontos correspondentes no objeto em suas novas posições. O Rastreamento de Pontos ocorre no

Figura 2.7: Supervisão de Movimentos e Rastreamento de Pontos: A partir de um vetor de características w específico, gera-se uma imagem e seus respectivos mapas de características. O usuário, então, insere na imagem os pares de pontos de manipulação (indicados em vermelho) e os pontos alvo (indicados em azul). Utiliza-se o otimizador Adam para derivar o novo vetor de características w', baseando-se na função de perda calculada conforme a Equação 2.2. Com o vetor w', produz-se uma nova imagem, seguida pela realização do Rastreamento de Pontos no mesmo espaço de características, conforme descrito pela Equação 2.3.



Fonte: (PAN et al., 2023)

mesmo espaço de características, utilizando a técnica de pesquisa do vizinho mais próximo em um *patch* nesse espaço. Especificamente, o vetor de características do ponto de manipulação inicial é representado por $\mathbf{f_i} = \mathbf{F_0}(\mathbf{p_i})$. O *patch* em torno de $\mathbf{p_i}$ é representado como $\Omega_2(\mathbf{p_i}, r_2) = \{(x, y) \mid |x - x_{p,i}| < r_2, |y - y_{p,i}| < r_2 \}$. O ponto rastreado é obtido a partir da busca do vizinho $\mathbf{q_i}$ mais semelhante a $\mathbf{f_i}$ em $\Omega_2(\mathbf{p_i}, r_2)$:

$$\mathbf{p_i} := \underset{\mathbf{q_i} \in \Omega_2(\mathbf{p_i}, r_2)}{\arg\min} \|\mathbf{F}'(\mathbf{q_i}) - \mathbf{f_i}\|_1. \tag{2.3}$$

Dessa forma, $\mathbf{p_i}$ é atualizado para rastrear o objeto na imagem. Para mais de um ponto de manipulação, o mesmo processo é aplicado a cada ponto individualmente. Após a conclusão do Rastreamento de Pontos, a etapa de otimização é repetida com base nas novas posições dos pontos de manipulação e nos vetores de características correspondentes. Essa otimização é realizada iterativamente até que cada ponto de manipulação p_i alcance

o seu respectivo ponto alvo t_i, o que normalmente requer de 30 a 200 iterações, conforme constatado nos experimentos realizados. O usuário tem a flexibilidade de interromper o processo de otimização em qualquer momento durante as iterações. Após a conclusão da edição, o usuário pode inserir novos conjuntos de pontos de manipulação e alvo e continuar realizando edições adicionais até obter resultados satisfatórios.

A implementação da DragGAN foi realizada com base no *framework* PyTorch (PASZKE et al., 2017). O otimizador Adam (KINGMA; BA, 2014) foi empregado para otimizar o vetor latente w, com um tamanho de passo de $2\mathrm{e}{-3}$ para os conjuntos de dados FFHQ (KARRAS; LAINE; AILA, 2019), AFHQCat (CHOI et al., 2020) e LSUN Car (YU et al., 2015), e $1\mathrm{e}{-3}$ para os demais conjuntos de dados. Os hiperparâmetros foram definidos como $\lambda = 20$, $r_1 = 3$ e $r_2 = 12$. O processo de otimização é interrompido quando todos os pontos de manipulação estão a uma distância máxima de d *pixels* dos respectivos pontos alvo correspondentes, onde d é estabelecido como 1 pixel para até 5 pontos de manipulação, e 2 pixels para mais de 5 pontos de manipulação. Além disso, foi desenvolvida uma interface gráfica para suportar a manipulação interativa de imagens.

DragGAN foi avaliada utilizando a arquitetura StyleGAN2 (KARRAS et al., 2020), a qual foi pré-treinada em diferentes conjuntos de dados. Os conjuntos de dados utilizados para o pré-treinamento incluíram (a resolução da StyleGAN2 pré-treinada é mostrada entre parênteses para cada conjunto de dados correspondente): FFHQ (512) (KARRAS; LAINE; AILA, 2019), AFHQCat (512) (CHOI et al., 2020), SHHQ (512) (FU et al., 2022), LSUN Car (512) (YU et al., 2015), LSUN Cat (256) (YU et al., 2015), Landscapes HQ (256) (SKOROKHODOV; SOTNIKOV; ELHOSEINY, 2021), Microscope (512) (PINKNEY, 2020), bem como um conjunto de dados sintético a partir de (MOKADY et al., 2022) contendo as categorias Leão (512), Cão (1024) e Elefante (512). Essa escolha diversificada de conjuntos de dados permitiu treinar a StyleGAN2 com uma ampla variedade de imagens, abrangendo retratos de alta qualidade, imagens de animais e paisagens, bem como objetos específicos capturados por um microscópio. Esses conjuntos de dados pré-treinados foram utilizados como base para as manipulações de imagens realizadas na abordagem proposta.

2.4 Comparativo das Técnicas Apresentadas

Os métodos DragGAN (PAN et al., 2023) e UserControllableLT (ENDO, 2022) apresentam configurações semelhantes, porém possuem diferenças significativas. O mé-

todo UserControllableLT não permite a criação de máscaras para delimitar regiões móveis e fixas na imagem, porém permite ao usuário definir um conjunto fixo de pontos que permanecerão inalterados durante o processo de edição. No entanto, não oferece um controle preciso sobre os pontos, o que pode resultar em mudanças indesejadas nas imagens, pois muitas vezes o ponto de manipulação não alcança o seu ponto alvo correspondente. Por outro lado, o método DragGAN lida de forma mais eficaz com o controle de múltiplos pontos de origem, garantindo que cada ponto atinja o seu ponto de destino de maneira mais precisa. Isso possibilita uma manipulação de imagem mais diversificada e precisa. Vale ressaltar que o método UserControllableLT é mais rápido do que o DragGAN, porém seus resultados são menos precisos.

2.5 Limitações da Técnica DragGAN

Apesar dos avanços significativos apresentados pela técnica DragGAN na manipulação interativa de imagens, esta abordagem não está isenta de limitações. Estas questões, que são cruciais para a compreensão completa do potencial e das restrições da técnica, são discutidas a seguir.

Primeiramente, uma das limitações da DragGAN está relacionada à sua dependência do espaço latente da StyleGAN2. Embora este espaço ofereça um alto grau de controle e versatilidade, ele também possui suas próprias limitações intrínsecas. A manipulação dos vetores de características pode não ser sempre precisa, especialmente em cenários complexos ou quando se deseja fazer ajustes muito finos em características específicas. Isto pode resultar em alterações indesejadas em outras partes da imagem, que não estão sob a manipulação direta do usuário.

Outra limitação importante é a necessidade de um número relativamente alto de iterações para alcançar a posição desejada dos pontos de manipulação. Essa necessidade pode se tornar um obstáculo para usuários que buscam resultados imediatos. Além disso, o processo de alcançar a posição desejada dos pontos de manipulação muitas vezes é demorado e requer alto poder computacional, o que pode ser um desafio em sistemas com recursos limitados ou para operações que necessitam de respostas rápidas.

A precisão do Rastreamento de Pontos também possui limitações. Embora a DragGAN faça uso de uma abordagem de rastreamento eficaz, há situações em que o rastreamento pode falhar, especialmente em cenários com alta complexidade visual. Isso pode levar a resultados imprecisos ou a uma necessidade de correções manuais adicionais, o

que pode ser demorado e tecnicamente desafiador.

Adicionalmente, a técnica pode ser vulnerável a artefatos visuais e distorções, particularmente em imagens com padrões complexos ou texturas detalhadas. A manipulação dos vetores de características pode, em alguns casos, levar à geração de artefatos que comprometam a qualidade visual da imagem editada. Esta situação é mais evidente em imagens onde a fidelidade e a precisão dos detalhes são cruciais.

Por fim, a aplicabilidade da DragGAN em diferentes tipos de imagens e cenários ainda é uma área que requer mais exploração. Enquanto a técnica demonstra um desempenho notável em uma variedade de conjuntos de dados, sua eficácia em cenários fora do contexto das distribuições de treinamento ou em imagens com características únicas ainda precisa ser amplamente validada. A Figura 2.8 ilustra situações em que as manipulações geradas estão fora da distribuição de treinamento, resultando em distorções nas imagens, como um leão com a boca extremamente aberta e uma roda que foi ampliada de forma exagerada.

Figura 2.8: Distorções são geradas ao tentar manipular a imagem para abrir exageradamente a boca do leão e ampliar a roda de forma desproporcional, indicando manipulações fora da distribuição de treinamento.

Fonte: (PAN et al., 2023)

Essas limitações destacam áreas importantes para futuras pesquisas e desenvolvimento. A superação desses desafios não só melhorará a técnica DragGAN, mas também contribuirá significativamente para o avanço do campo da edição de imagens baseadas em modelos generativos.

3 ACELERAÇÃO DA TÉCNICA DRAGGAN

Esta seção detalha a metodologia adotada para aprimorar a técnica DragGAN, enfatizando a aceleração da manipulação de vetores de características. A Seção 3.1 introduz o método proposto para atualização dos vetores de características para obtenção das edições desejadas, baseando-se na análise das diferenças entre os vetores de características iniciais e os obtidos após as primeiras otimizações pelo método tradicional. A Seção 3.2 discorre sobre os ajustes específicos realizados nos parâmetros principais para otimizar o processo de edição. Na Seção 3.3, são descritas as tecnologias principais utilizadas no desenvolvimento e implementação do método. Finalmente, a Seção 3.4 apresenta a metodologia empregada na avaliação da eficiência do método proposto em comparação com a abordagem tradicional, com foco nas melhorias em termos de tempo de processamento e redução no número de iterações.

3.1 Método Proposto para a Otimização de Vetores de Características

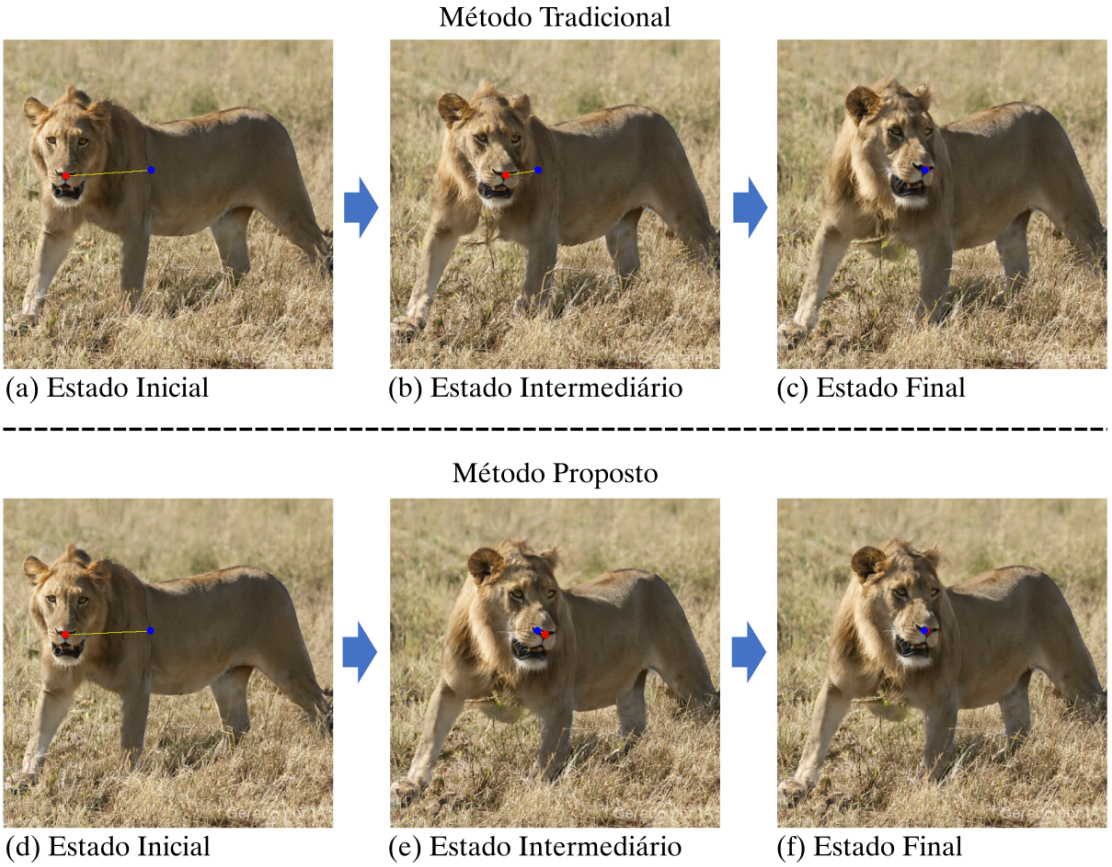
O foco principal deste trabalho é a proposta de um método alternativo para acelerar a otimização de vetores de características, visando uma maior rapidez em comparação com o procedimento convencional. A nova abordagem baseia-se na análise das diferenças entre os vetores de características iniciais e aqueles obtidos após os três primeiros passos de otimização utilizando o método tradicional.

3.1.1 Processo de Otimização Tradicional Revisitado

Inicialmente, um vetor de características arbitrário w é alimentado na rede geradora da StyleGAN2, gerando uma imagem I. O usuário então define pares de pontos de manipulação (vermelho) e alvo (azul), representando as alterações desejadas na imagem, conforme ilustrado na Figura 3.1 (a). Após esta definição, inicia-se a sequência de iterações para aproximar cada ponto de manipulação p_i do seu respectivo alvo t_i. Conforme descrito na Seção 2.3, este processo divide-se em Supervisão de Movimentos e Rastreamento de Pontos. A primeira etapa, a Supervisão de Movimentos, é onde se dá a otimização do vetor w através do otimizador Adam, enquanto os pesos da StyleGAN2 permanecem inalterados. Este vetor atualizado é então utilizado na rede geradora da

StyleGAN2 para produzir uma nova imagem I' com mudanças incrementais. O Rastreamento de Pontos segue, ajustando a posição de cada ponto de manipulação na imagem I', e o ciclo se repete até que cada ponto alcance seu alvo, conforme ilustrado na Figura 3.1 (c).

Figura 3.1: Exemplo de edição de imagem utilizando os métodos tradicional (acima) e proposto (abaixo). Em cada imagem, o ponto vermelho representa um ponto de manipulação, enquanto o ponto azul indica um ponto alvo. As imagens (a) e (d) mostram os estados iniciais, as imagens (b) e (e) exibem estados intermediários durante as iterações, e as imagens (c) e (f) ilustram a conclusão das iterações.



Fonte: O Autor

3.1.2 Processo de Aceleração Proposto

O objetivo da técnica proposta é acelerar o processo de edição, reduzindo o número de iterações que utilizam o otimizador, visto que este processo é computacionalmente intensivo. Ao invés disso, estima-se a direção no espaço latente da StyleGAN2 que atualiza o vetor de características w de modo a levar a vizinhança $\Omega(\mathbf{p_i}, r_2)$ centrada no ponto inicial de manipulação $\mathbf{p_i}$ para a uma nova região no espaço de características

centrada no ponto alvo $\mathbf{t_i}$.

Inicialmente, o processo começa com o salvamento de uma cópia do vetor de características inicial, w_0. Em seguida, realiza-se três iterações tradicionais utilizando o otimizador Adam, obtendo assim um valor atualizado do vetor w. Após essas iterações, obtém-se a diferença $\Delta w = w - w_0$ entre o vetor de características atualizado e o inicial. A escolha de manter o número total de iterações otimizadas em três baseia-se em observações experimentais. Esta quantidade mostrou-se mais eficaz para a maioria dos casos. Visto que o incremento em uma única iteração tende a ser muito pequeno, o uso de apenas uma ou duas iterações tende a resultar em Δw com algumas componentes nulas (em virtude de erros de aproximação), ocultando informações relevantes para as otimizações. Por outro lado, um número excessivo de iterações poderia comprometer a eficiência do método.

Com as três iterações iniciais estabelecendo uma base para o refinamento subsequente, o método proposto é então empregado nas iterações seguintes. Nesta fase, ao invés de usar o otimizador Adam, realiza-se diretamente a soma do vetor Δw obtido anteriormente ao vetor de características atual, w. Esta modificação aplica-se apenas à etapa de Supervisão de Movimentos, mantendo inalterada a etapa de Rastreamento de Pontos.

Este processo é aplicado continuamente até que a distância entre os pontos de manipulação e os alvos deixe de diminuir, o que é ilustrado na Figura 3.1 (e). Quando um aumento nesta distância é observado, verifica-se a necessidade de um ajuste mais refinado. Para tal, o otimizador Adam é utilizado nas últimas iterações, permitindo um ajuste fino dos vetores de características. Variáveis de controle são empregadas para monitorar a variação das distâncias entre os pontos de manipulação e alvos ao longo das iterações. O resultado final obtido com o método proposto, como visto na Figura 3.1 (f), é comparável ao alcançado pelo método tradicional, demonstrado na Figura 3.1 (c). Após a conclusão de uma etapa de edição, o usuário pode definir novos pontos de manipulação e alvo, repetindo o processo conforme necessário.

Adicionalmente, o valor de Δw pode ser escalado por uma constante $a > 1$ com a finalidade de aumentar o tamanho do passo e acelerar a convergência de cada ponto de manipulação em direção ao seu alvo respectivo. Este procedimento deve ser conduzido com cautela, pois um passo excessivamente grande pode afetar negativamente a etapa de Rastreamento de Pontos subsequente. Se o Rastreamento de Pontos não operar corretamente, os pontos de manipulação podem ser rastreados de maneira incorreta, resultando em alterações indesejadas na imagem.

3.2 Parâmetros de Otimização

Para acelerar o processo de edição, dois parâmetros principais foram ajustados:

- ***Learning Rate***: O *learning rate* foi aumentado de 0.001 para 0.002. Este ajuste visa acelerar a convergência da otimização tradicional, tornando as modificações nos vetores de características mais rápidas e eficientes. Experimentos foram conduzidos com *learning rates* variando de 0.001 a 0.007, e o valor de 0.002 demonstrou o melhor equilíbrio entre desempenho e precisão. Valores mais altos de *learning rate* podem comprometer a etapa subsequente de Rastreamento de Pontos, uma vez que passos muito grandes podem resultar em mudanças drásticas na imagem, exigindo um aumento excessivo do raio do Rastreamento de Pontos (r2). Isso aumentaria o risco de selecionar pontos incorretos durante o rastreamento, devido à abrangência maior da área de busca.
- **Raio de Rastreamento de Pontos (r2):** O raio de Rastreamento de Pontos foi expandido de 12 para 36 pixels. Esse aumento é necessário para adaptar-se aos passos maiores na otimização introduzidos pelo uso do incremento Δw, proporcionando um rastreamento mais eficaz e abrangente dos pontos de manipulação. Tal ajuste é crucial para garantir uma edição mais precisa e detalhada da imagem, especialmente considerando o aumento do *learning rate* e a aplicação das otimizações alternativas. A definição do raio em 36 foi baseada em uma série de experimentos, sendo este valor o mais eficiente e preciso para os casos testados.

3.3 Tecnologias Utilizadas

Este trabalho foi desenvolvido com base na implementação original da DragGAN, realizando modificações sobre o *framework* original desenvolvido pelos autores. Para estas alterações, utilizamos a linguagem Python e o *framework* de aprendizado de máquina PyTorch, amplamente empregado na comunidade científica para manipulação de tensores e redes neurais. Devido ao elevado poder computacional exigido pelo processo, empregamos o Google Colab (versão gratuita), uma plataforma de notebooks Jupyter que oferece GPUs e um ambiente de desenvolvimento baseado na nuvem. Esta escolha acelerou significativamente os cálculos necessários para a otimização dos vetores de características, especialmente diante da complexidade computacional das operações em redes neurais

profundas. A interface gráfica de usuário foi baseada na biblioteca Gradio, uma ferramenta *open-source* que permite a criação rápida de interfaces *web* para a demonstração de aplicações de aprendizado de máquina. Por fim, a medição do tempo de processamento ocorreu por meio da biblioteca `time` de Python.

3.4 Avaliação da Eficiência do Método

A eficácia do método proposto foi avaliada em comparação com a abordagem tradicional, evidenciando maior eficiência tanto em termos de tempo de processamento, como consequência da redução do número de iterações do método de otimização necessárias durante o processo de edição. A medição do tempo iniciou-se imediatamente após o usuário emitir o comando para iniciar as iterações de otimização, sendo concluída quando todos os pontos de manipulação alcançaram seus alvos, marcando o término da otimização. Cada experimento foi realizado dez vezes, e os valores apresentados são as médias aritméticas dessas execuções. Adicionalmente, registrou-se o número total de iterações desde o início do processo, possibilitando uma comparação com o método tradicional. Para o método proposto, foram contabilizados os totais de iterações utilizando tanto as otimizações tradicionais quanto as iterações propostas, permitindo avaliar o impacto das otimizações propostas no tempo total de edição. Avaliações qualitativas das imagens geradas também foram realizadas. As discussões e os resultados referentes aos experimentos são abordados no Capítulo 4.

4 RESULTADOS

Este capítulo apresenta os resultados alcançados com a implementação das modificações propostas para a técnica DragGAN original e discute os impactos dessas alterações. A avaliação concentra-se em comparar a eficiência e a qualidade dos resultados obtidos com o método modificado em relação ao método tradicional. Além disso, este capítulo aborda os desafios enfrentados durante este estudo e as limitações identificadas, propondo caminhos para futuras investigações.

4.1 Avaliação Quantitativa da Solução Proposta

A implementação da solução proposta para otimização de vetores de características resultou em melhorias na eficiência do processo de edição de imagens. Em comparação com o método tradicional, que utiliza exclusivamente o otimizador Adam, observou-se uma redução média no tempo de processamento por imagem de aproximadamente 22% nos casos envolvendo um único par de pontos, conforme exemplificado na Tabela 4.1. Nos casos com dois pares de pontos, mencionados na Tabela 4.2, a redução média no tempo de processamento foi de aproximadamente 8%. Adicionalmente, a quantidade total de iterações necessárias durante o processo de manipulação diminuiu em média cerca de 18% para os experimentos com um par de pontos e 8% para aqueles com dois pares de pontos. Estes resultados sugerem que a combinação das otimizações iniciais utilizando o otimizador Adam, seguida pela aplicação sucessiva do incremento Δw no espaço latente, e complementada por um refinamento novamente utilizando o otimizador Adam oferece ganhos de desempenho significativos.

4.2 Avaliação Qualitativa da Solução Proposta

Além da eficiência, avaliou-se de forma empírica a qualidade das imagens editadas. Por meio de comparações visuais entre as imagens geradas pelos métodos tradicional e proposto, conforme ilustrado nas Figuras 4.2 a 4.9, observou-se que o método proposto preservou um alto grau de fidelidade visual. Os resultados alcançados foram muito semelhantes aos obtidos com o método original. Não foram identificados artefatos significativos nem distorções que pudessem comprometer a qualidade das imagens. Esse achado

Tabela 4.1: Comparação entre os tempos de execução e o número de iterações dos métodos tradicional e proposto em experimentos de edição utilizando um único par de pontos de manipulação. Cada experimento foi repetido dez vezes, e os valores apresentados nesta tabela representam as médias dessas execuções. No cabeçalho, D denota a distância média, em *pixels*, entre os pontos de manipulação e os alvos; I_t é a média do total de iterações com o método tradicional; I_p indica a média do total de iterações com o método proposto; T_p refere-se ao tempo médio de processamento em segundos; e R_{T_p} mostra a porcentagem de redução de tempo de processamento obtida com o método proposto.

			Método Tradicional		**Método Proposto**			
Fig.	*Experimento*	D	I_t	T_p (s)	I_t	I_p	T_p (s)	R_{T_p} (%)
4.2	Cão	164,20	26,40	29,63	9,90	9,00	20,60	30,48
4.3	Elefante 1	30,41	22,00	8,04	9,20	10,50	7,01	12,81
4.4	Leão 1	56,56	32,70	11,89	6,10	13,20	6,63	44,24
4.5	Leão 2	124,47	34,90	13,44	15,40	19,00	12,46	7,29
4.6	Cavalo 1	35,02	49,40	7,20	31,10	13,40	6,01	16,53

Fonte: O Autor

indica que a implementação do método proposto, além de acelerar o processo de edição, manteve a integridade visual das imagens resultantes.

4.3 Discussão

Durante a implementação das modificações na técnica DragGAN, enfrentamos vários desafios, especialmente na calibração do novo método de otimização. O ajuste cuidadoso do *learning rate* e do raio de rastreamento de pontos foi crucial para atingir um equilíbrio entre eficiência e precisão. Realizamos uma série de experimentos para determinar os parâmetros ideais. Valores excessivamente altos do *learning rate* poderiam acelerar demais o processo, comprometendo a precisão do rastreamento de pontos, uma vez que um passo maior do que a abrangência do raio de rastreamento impediria o rastreamento correto dos novos pontos. Por outro lado, o uso de um raio de rastreamento demasiadamente amplo poderia levar ao rastreamento de pontos incorretos e afetar negativamente o tempo de processamento. Integrar harmoniosamente o método proposto de otimização de vetores de características com o método tradicional, sem prejudicar a qualidade dos resultados finais e ainda alcançando melhorias de eficiência, constituiu um desafio.

Observamos que a técnica proposta apresenta desempenho superior à técnica convencional com um par de pontos em comparação com dois ou mais pares. Os experi-

Tabela 4.2: Comparação entre os tempos de execução e o número de iterações dos métodos tradicional e proposto em experimentos com múltiplos pares de pontos de manipulação. Cada experimento foi realizado dez vezes, e os valores nesta tabela são as médias dessas execuções. No cabeçalho, D_1 e D_2 representam as distâncias médias, em *pixels*, entre os primeiros e segundos pares de pontos de manipulação e seus alvos, respectivamente; I_t é a média do total de iterações com o método tradicional; I_p indica a média do total de iterações com o método proposto; T_p refere-se ao tempo médio de processamento em segundos; e R_{T_p} mostra a porcentagem de redução de tempo de processamento alcançada com o método proposto.

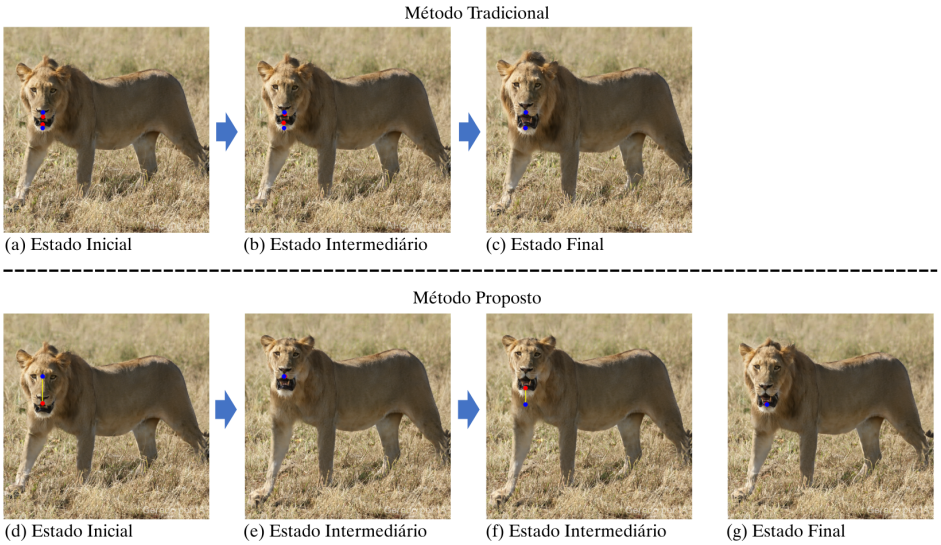
				Tradicional		**Método Proposto**			R_{T_p}
Fig.	*Experimento*	D_1	D_2	I_t	T_p (s)	I_t	I_p	T_p (s)	(%)
4.7	Cavalo 2	20,47	20,63	99,40	13,97	92,00	2,20	13,40	4,08
4.8	Carro	28,54	17,72	186,70	56,30	162,50	24,10	55,85	0,80
4.9	Elefante 2	22,68	15,70	41,90	15,70	32,90	0,90	12,49	20,45

Fonte: O Autor

mentos indicaram que, devido à dinâmica das otimizações, frequentemente um ou ambos os pares de pontos experimentam um aumento sutil na distância entre seus pontos de manipulação e alvo durante algumas iterações de otimização, levando à adoção da metodologia tradicional. Assim, ao usar dois ou mais pares de pontos, os resultados tendem a ser similares aos obtidos com a técnica tradicional. No entanto, muitas operações que envolvem dois pares de pontos podem ser desmembradas em operações mais simples, cada uma especificada com um par de pontos, obtendo resultados equivalentes e aproveitando os benefícios da técnica proposta. A Figura 4.1 ilustra como uma edição, originalmente realizada com dois pares de pontos pela técnica tradicional (Figuras a, b e c), pode ser substituída por duas edições, cada uma realizada com um único par de pontos, utilizando a técnica proposta (Figuras d, e, f e g).

Os experimentos representados pelas Figuras 4.2, 4.3 e 4.4 foram reutilizados para avaliar o impacto do aumento no *learning rate* e do método proposto de otimização de vetores latentes no desempenho do tempo de processamento. Na comparação dos tempos médios de processamento, observou-se um aumento de 1,03 segundos ao se elevar o *learning rate* de 0.001 para 0.002 com o método proposto. Por outro lado, ao comparar o desempenho do método proposto, mantendo o *learning rate* em 0.001, com o método original também a 0.001, constatou-se que o original levou, em média, 5,11 segundos a mais. Tal diferença reforça que o ganho de desempenho deve-se primordialmente ao método proposto de otimização de vetores latentes, mais do que ao aumento do *learning rate*, evidenciando a eficácia deste novo método em aprimorar a eficiência do processo sem prejudicar a qualidade dos resultados.

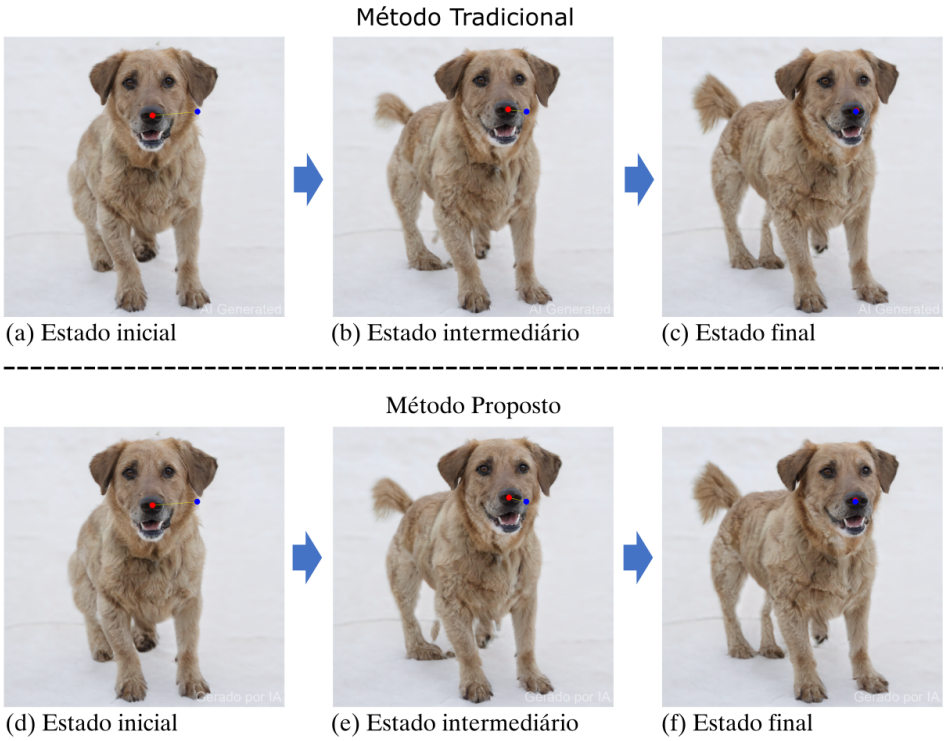
Figura 4.1: Na edição de imagem apresentada em (a), (b) e (c), utiliza-se a técnica tradicional para fazer o leão abrir a boca, empregando dois pares de pontos numa única etapa. Em contraste, as imagens (d), (e), (f) e (g) ilustram a utilização de um único par de pontos para efetuar a mesma edição, mas dividida em duas etapas distintas. Os pares de pontos de manipulação e alvo são estabelecidos em (d) e (f), enquanto (e) e (g) exibem os resultados intermédio e final da edição, respectivamente.



Fonte: O Autor

As modificações introduzidas na técnica DragGAN original, particularmente a utilização de uma alternativa simplificada ao método de otimização, representam um modesto avanço no campo da edição de imagens baseadas em StyleGANs. Essas alterações não apenas aumentaram a eficiência do processo, mas também mantiveram a qualidade das imagens editadas. Este equilíbrio entre velocidade e qualidade é essencial para a aplicabilidade prática da técnica em cenários reais. Baseado nos resultados obtidos, vislumbramos um vasto campo para futuras pesquisas e aplicações. Estudos subsequentes podem focar na otimização da eficiência do método proposto, na exploração de seu uso em diferentes tipos de GANs e na avaliação de sua aplicabilidade em contextos mais amplos de edição de imagens.

Figura 4.2: Experimento de manipulação de um par de pontos: Movimentação lateral da posição do cão. Pontos de manipulação indicados em vermelho e pontos alvo em azul.

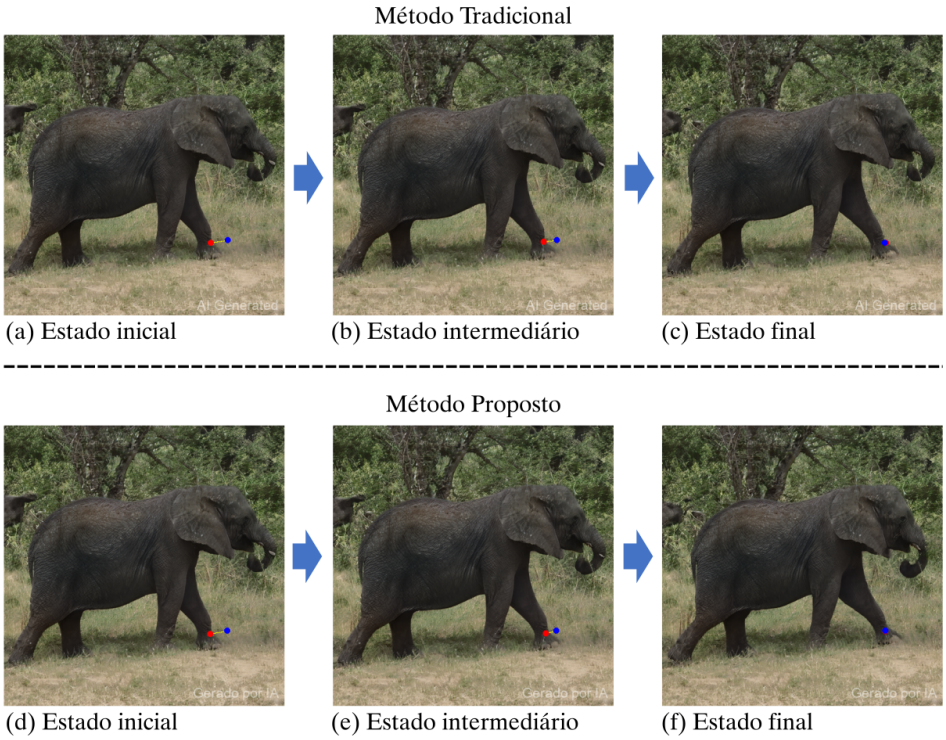


(a) Estado inicial (b) Estado intermediário (c) Estado final

(d) Estado inicial (e) Estado intermediário (f) Estado final

Fonte: O Autor

Figura 4.3: Experimento de manipulação de um par de pontos: Movimentação da pata dianteira esquerda do elefante para frente. Pontos de manipulação indicados em vermelho e pontos alvo em azul.

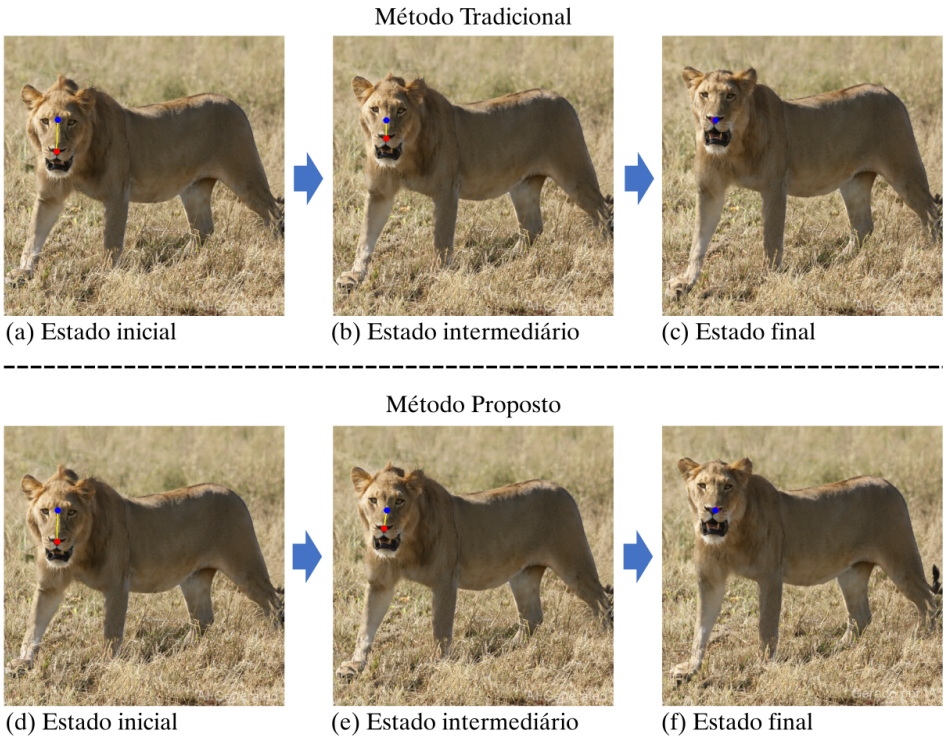


(a) Estado inicial (b) Estado intermediário (c) Estado final

(d) Estado inicial (e) Estado intermediário (f) Estado final

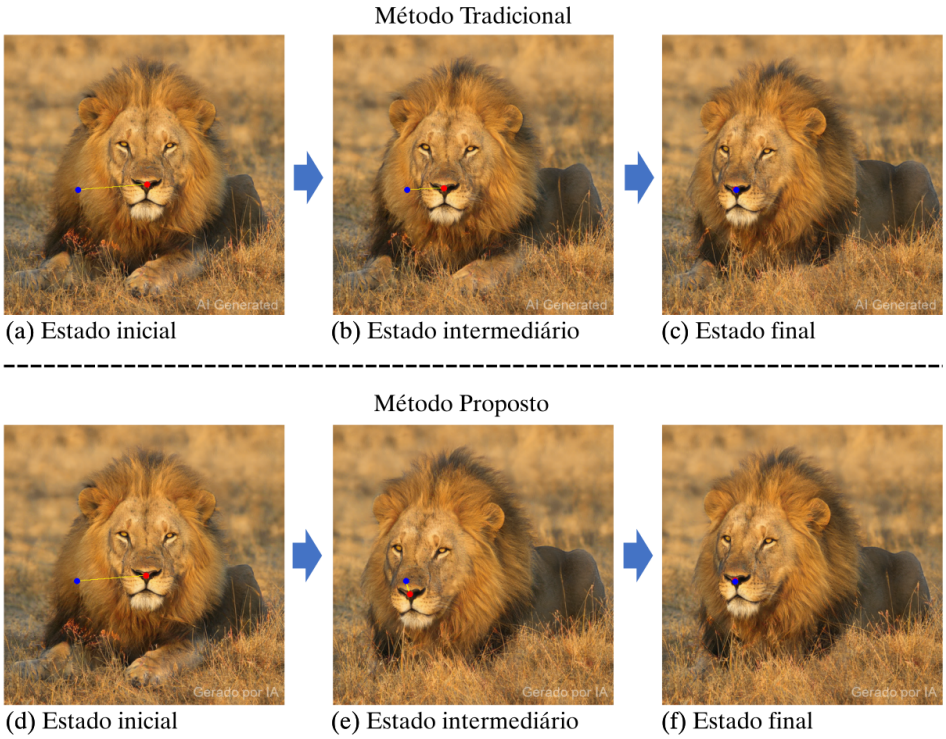
Fonte: O Autor

Figura 4.4: Experimento de manipulação de um par de pontos: Elevação do focinho da leoa. Pontos de manipulação indicados em vermelho e pontos alvo em azul.



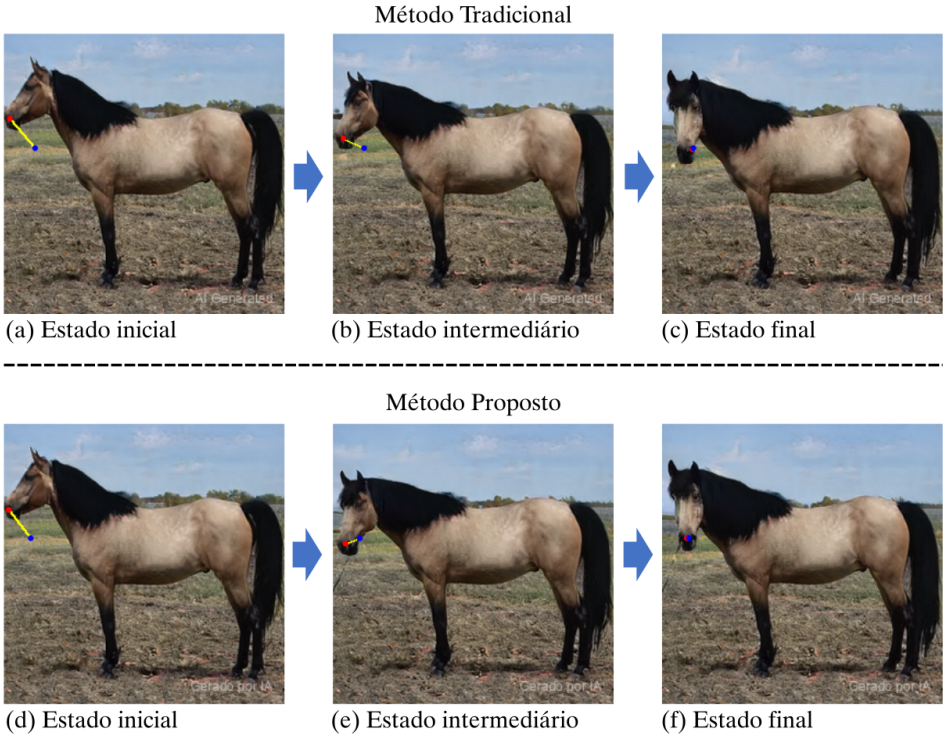
Fonte: O Autor

Figura 4.5: Experimento de manipulação de um par de pontos: Deslocamento lateral do focinho do leão. Pontos de manipulação indicados em vermelho e pontos alvo em azul.



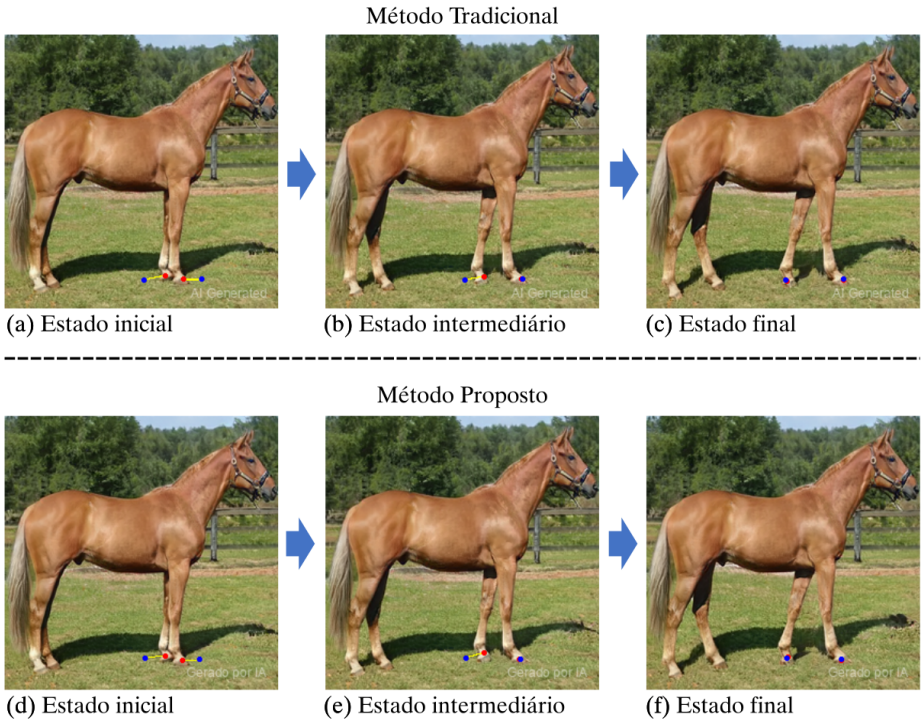
Fonte: O Autor

Figura 4.6: Experimento de manipulação de um par de pontos: Rotacionamento da cabeça do cavalo. Pontos de manipulação indicados em vermelho e pontos alvo em azul.



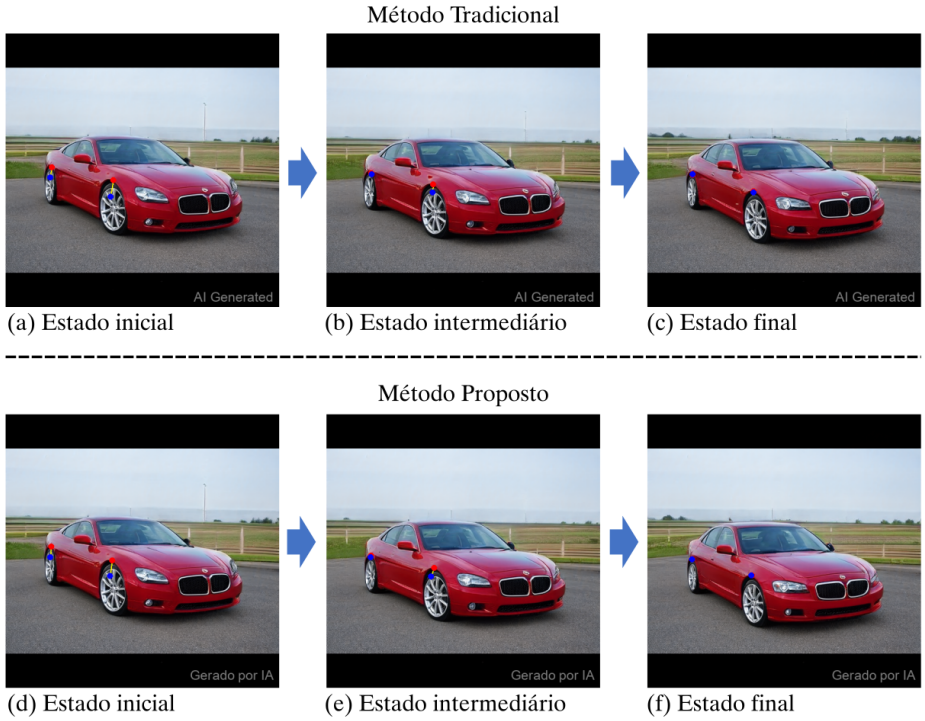
Fonte: O Autor

Figura 4.7: Experimento de manipulação de dois pares de pontos: Separação das patas dianteiras do cavalo. Pontos de manipulação indicados em vermelho e pontos alvo em azul.



Fonte: O Autor

Figura 4.8: Experimento de manipulação de dois pares de pontos: Redimensionamento das rodas e reajuste posicional de um veículo. Pontos de manipulação em vermelho e pontos alvo em azul.

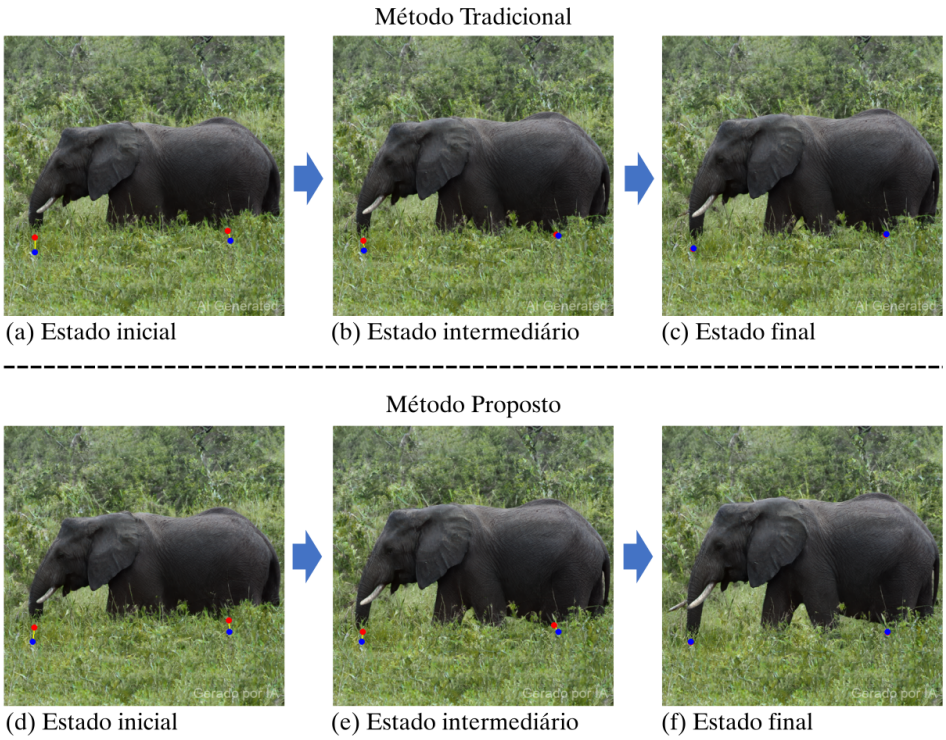


(a) Estado inicial (b) Estado intermediário (c) Estado final

(d) Estado inicial (e) Estado intermediário (f) Estado final

Fonte: O Autor

Figura 4.9: Experimento de manipulação de dois pares de pontos: Diminuição da vegetação sob o elefante. Pontos de manipulação em vermelho e pontos alvo em azul.



(a) Estado inicial (b) Estado intermediário (c) Estado final

(d) Estado inicial (e) Estado intermediário (f) Estado final

Fonte: O Autor

5 CONCLUSÕES E FUTUROS TRABALHOS

Este trabalho introduziu um método para acelerar o processo de edição de imagens baseadas em StyleGANs, com foco na técnica DragGAN. Desenvolveu-se um método alternativo de otimização de vetores de características, combinando as otimizações iniciais com o método Adam e um novo método baseado na soma de vetores no espaço latente. Realizaram-se também ajustes em parâmetros como o *learning rate* e o raio de rastreamento de pontos, visando aprimorar a eficiência do processo. Os resultados demonstraram melhorias na eficiência da edição de imagens, evidenciada pela redução no tempo médio de processamento e no número total de iterações necessárias. A qualidade das imagens editadas foi preservada, sem a ocorrência de artefatos significativos ou distorções.

Alternativas para pesquisas futuras incluem estudos para alcançar eficiência ainda maior no processo de otimização, testar sua aplicabilidade em diferentes tipos de GANs para expandir seu escopo de uso, e avaliar sua eficácia em um leque mais amplo de cenários de edição de imagens. Investigar abordagens para aprimorar o controle sobre múltiplos pontos de manipulação e minimizar artefatos visuais e distorções, especialmente em imagens com padrões complexos ou texturas detalhadas, é fundamental. Outra linha de trabalho futura pode envolver o desenvolvimento de termos de regularização para manter as imagens geradas durante a edição dentro da distribuição de treinamento, minimizando artefatos indesejados.

REFERÊNCIAS

CHOI, Y. et al. Stargan v2: Diverse image synthesis for multiple domains. In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2020.

ENDO, Y. User-controllable latent transformer for stylegan image layout editing. **Computer Graphics Forum**, v. 41, n. 7, p. 395–406, 2022.

FU, J. et al. Stylegan-human: A data-centric odyssey of human generation. In: **European Conference on Computer Vision (ECCV)**. [S.l.: s.n.], 2022.

GOODFELLOW, I. et al. Generative adversarial nets. In: **Advances in Neural Information Processing Systems (NeurIPS)**. [S.l.: s.n.], 2014.

KARRAS, T.; LAINE, S.; AILA, T. A style-based generator architecture for generative adversarial networks. In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2019.

KARRAS, T. et al. Analyzing and improving the image quality of stylegan. In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2020.

KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. **arXiv preprint arXiv:1412.6980**, 2014.

MOKADY, R. et al. Self-distilled stylegan: Towards generation from internet photos. In: **ACM SIGGRAPH 2022 Conference Proceedings**. [S.l.: s.n.], 2022. p. 1–9.

PAN, X. et al. Drag your gan: Interactive point-based manipulation on the generative image manifold. In: **ACM SIGGRAPH 2023 Conference Proceedings**. [S.l.: s.n.], 2023.

PASZKE, A. et al. **Automatic differentiation in PyTorch**. 2017.

PINKNEY, J. N. M. **Awesome pretrained StyleGAN2**. 2020. Https://github.com/justinpinkney/awesome-pretrained-stylegan2.

SKOROKHODOV, I.; SOTNIKOV, G.; ELHOSEINY, M. **Aligning Latent and Image Spaces to Connect the Unconnectable**. 2021. ArXiv preprint arXiv:2104.06954.

YU, F. et al. **LSUN: Construction of a Large-scale Image Dataset using Deep Learning with Humans in the Loop**. 2015. ArXiv preprint arXiv:1506.03365.