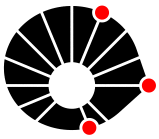
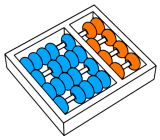




Aumento de dados com modelos de difusão *image to image* e GANs para melhoria na generalização de detectores de deepfake

Guilherme Pereira Corrêa *Esther Luna Colombini*

Relatório Técnico - IC-PFG-23-08
Projeto Final de Graduação
2023 - Julho

UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE COMPUTAÇÃO

The contents of this report are the sole responsibility of the authors.
O conteúdo deste relatório é de única responsabilidade dos autores.

Aumento de dados com modelos de difusão *image to image* e GANs para melhoria na generalização de detectores de deepfake

Guilherme Pereira Corrêa Esther Luna Colombini*

Resumo

Deepfakes são mídias sintéticas geradas por redes neurais que podem apresentar conteúdo prejudicial à sociedade, como notícias falsas e fraudes. Considerando a evolução constante de tais tecnologias, é importante desenvolver métodos que possam detectá-las automaticamente e uma das formas mais eficazes de se fazer isso é com modelos de aprendizado de máquina. No entanto, o desempenho de tais detectores depende do contexto dos dados de treinamento e pode diminuir consideravelmente quando aplicados a testes em contextos não vistos durante o treino. Na detecção de deepfakes isso é especialmente desafiador, pois as técnicas de geração estão em constante evolução, implicando que tais modelos apresentem uma queda de desempenho no mundo real. Neste projeto, exploramos o impacto do aumento de dados utilizando modelos de difusão "image to image" e Redes Generativas Adversariais (GANs) na capacidade de generalização de detectores de deepfake para domínios não observados no treinamento, realizando testes intra e cross dataset para uma avaliação mais abrangente dos resultados.

1 Introdução

A detecção de deepfake tem se tornado um tema de crescente relevância no campo da visão computacional, principalmente devido ao aumento significativo na disseminação de conteúdos falsos e manipulados usando essa tecnologia. Deepfakes são mídias sintéticas que apresentam uma aparência realista, mas na verdade são geradas por redes neurais profundas [1].

Embora o termo originalmente se refira a mídias em geral (imagens, vídeos ou áudios) que são difíceis de serem percebidos como falsos, sua popularização ocorreu principalmente no contexto de manipulação facial. Dado este contexto, deepfake pode ser categorizada em quatro grupos principais de acordo com sua funcionalidade, independentemente do modelo utilizado para geração. São eles: troca de identidade, troca de expressão, manipulação de atributos faciais e síntese de rosto completo. Neste estudo, o foco está na detecção de deepfakes de troca de identidade, mais especificamente em modelos que operam em nível de imagem. Além disso, foca-se em detecção com o uso de redes neurais, uma classe de modelos de aprendizado de máquina que demonstraram excelentes resultados em tarefas de visão computacional. Esses modelos têm a capacidade de aprender características e padrões complexos nos dados de entrada, o que os torna ideais para identificar indícios sutis de manipulação em conteúdos multimídia.

No entanto, o bom desempenho desses modelos geralmente está relacionado ao contexto dos dados nos quais foram treinados. Muitas vezes, as métricas de desempenho pioram significativamente quando o contexto dos dados de teste é diferente, mesmo que a tarefa seja a mesma. No caso da detecção de deepfakes, essa discrepância é ainda mais comum, uma vez que as tecnologias para geração de mídias falsas estão em constante evolução. Isso significa que modelos que não foram

*Instituto de Computação, Universidade Estadual de Campinas, 13081-970 Campinas, SP.

desenvolvidos levando em conta essa evolução tendem a apresentar queda de desempenho no mundo real.

Para melhorar a capacidade de generalização dos modelos nesse aspecto, diversas abordagens têm sido exploradas, como pré-processamento, aumento de dados (data augmentation) e treinamento de classificadores de uma única classe. Neste trabalho, investiga-se o impacto do aumento de dados utilizando grandes modelos geradores de imagem, que têm desempenhado um papel revolucionário nessa área. Mais especificamente, direcionamos nossa atenção para duas classes de modelos específicos: modelos de difusão *image to image* (img2img), como o DALL-E 2 [2], e modelos geradores de imagens baseados em redes neurais generativas adversariais (GANs), como a StyleGAN [3]. Além disso, para auxiliar na produção dos prompts para o modelo img2img, empregou-se um modelo de difusão *text to text*(txt2txt), evitando a necessidade de criar-se tais instruções manualmente, o que seria altamente custoso em termos de esforço e tempo.

1.1 Objetivo

Considerando o contexto apresentado, o objetivo deste trabalho é avaliar como o aumento de dados afeta a capacidade do modelo de detectar imagens de faces falsas cujo contexto não foi visto durante o treinamento, investigando cenários intra dataset, nos quais o domínio dos dados de teste é o mesmo do treinamento, e cenários cross dataset, nos quais os dados de teste são provenientes de conjuntos de dados não utilizados durante o treinamento.

Por meio desses experimentos, almejamos obter insights sobre o impacto do aumento de dados no desempenho da detecção de deepfakes em diferentes contextos e contribuir para o avanço das técnicas de detecção nesse cenário em constante evolução. A ameaça representada pelos deepfakes para a sociedade torna essas pesquisas e o desenvolvimento contínuo de modelos de detecção de extrema importância, visando fortalecer a segurança e a confiança nas mídias digitais, preservando a integridade das informações e mitigando os riscos associados aos deepfakes.

2 Revisão Bibliográfica

Conforme mencionado anteriormente, no âmbito dos deepfakes de imagens faciais, é comum classificá-los em quatro grupos distintos. Embora haja técnicas comuns empregadas na geração desses deepfakes, algumas delas são mais apropriadas para determinados grupos do que para outros. Esses grupos estão descritos a seguir e podem ser visualizados na Figura 1.

- Troca de identidade (*identity swap*)

  A troca de identidade envolve substituir a face de uma pessoa em uma imagem ou vídeo alvo pela face de uma pessoa em uma imagem ou vídeo fonte [4], mantendo o restante do corpo e do ambiente inalterados, é o tipo mais popular de deepfake em ataque nocivos, como em troca de identidade em vídeos pornográficos. Existem diversas técnicas disponíveis para realizar essa tarefa; por exemplo, Korshunova et al. [5] usam redes neurais convolucionais para fazer a troca de faces, enquanto Nirkin et al. [6] utilizam a StyleGAN2 [7] juntamente com uma representação em megapixels.

- Troca de expressão (*expression swap*)

  Este tipo de deepfake envolve substituir a expressão facial de uma pessoa na imagem ou vídeo alvo pela expressão facial de outra pessoa na imagem ou vídeo fonte. Doukas et al. [8] utilizam GANs para manipular a expressão facial em imagens faciais, enquanto Cao et al. [9]

propõem um framework de edição de vídeos mais consistente temporalmente, usando GANs sem especificar uma tarefa específica.

- Manipulação de atributos faciais (*attribute manipulation*)

  A manipulação de atributos faciais consiste na modificação de características do rosto, como cor de cabelo, cor de pele, gênero, adição de óculos, entre outros. Esse tipo de técnica de deepfake tornou-se muito popular nos últimos anos, especialmente em 2019 com o sucesso do aplicativo FaceApp. Para realizar essa tarefa, por exemplo, são utilizadas técnicas baseadas em GANs guiadas por máscaras e modelos de face 3D, como descrito em [10] e [11], respectivamente.

- Síntese de rosto completo (*entire face synthesis*)

  A síntese de rosto completo consiste em gerar novas faces inteiras, ou seja, criar faces de pessoas que não existem. Websites como o ThisPersonDoesNotExist.com ficaram muito famosos por surpreender as pessoas com rostos extremamente realistas de pessoas que nunca existiram. Nessa tarefa, as GANs são, de longe, o método mais utilizado e, em especial, a StyleGAN.

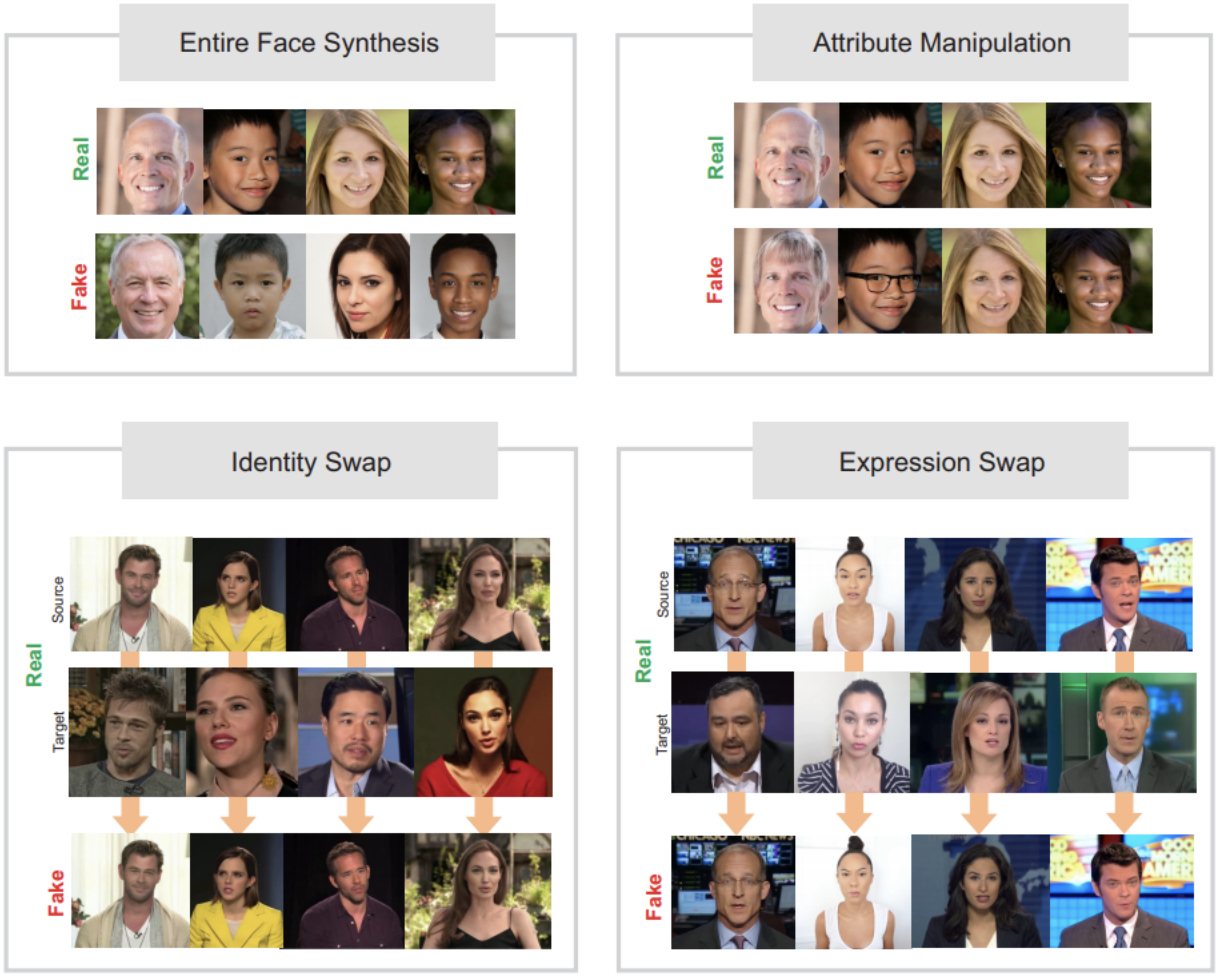


Figura 1: Exemplos de imagens reais e falsas para os quatro tipos diferentes de deepfake citados.

Um dos aspectos mais interessantes da StyleGAN é o seu espaço latente aprendido, que, apesar de ser obtido por meio de treinamento não supervisionado, apresenta um comportamento surpreendentemente bem comportado [12]. No contexto da StyleGAN para geração de imagens faciais, o espaço latente refere-se a um espaço vetorial de alta dimensão que representa as faces de uma maneira não interpretável por humanos, mas que o gerador do modelo é capaz de transformar em imagens. A ideia subjacente é que faces com características semelhantes estejam próximas umas das outras nesse espaço, permitindo a exploração de diferentes regiões para controlar diversos atributos ou características das imagens geradas, como identidade, idade, pose, expressão e cor do cabelo.

No entanto, os vetores nesse espaço são construídos globalmente, o que implica que diferentes atributos podem estar correlacionados, levando a alterações não esperadas em atributos ou áreas ao editar determinados aspectos. Para contornar essa limitação, Shi et al. propuseram a SemanticStyleGAN [13], no qual o gerador é treinado para modelar partes semânticas locais da face de forma separada e, posteriormente, sintetizá-las em uma face composta. Essa arquitetura tornou-se muito popular e pode ser utilizada para gerar deepfakes nos quatro grupos mencionados anteriormente, além de contar com código aberto. Neste trabalho, a rede SemanticStyleGAN foi empregada na tarefa de síntese de rostos completos, a fim de gerar mais imagens faciais falsas e expandir o conjunto de treinamento.

Assim como qualquer outra tarefa de classificação de imagens, a detecção de deepfakes pode ser feita seguindo métodos convencionais de aprendizado, que são baseadas totalmente em features criadas manualmente, ou em abordagens com aprendizado profundo, que tem foco em features aprendidas. Dado que a detecção de deepfake baseada em aprendizado tem vantagens na generalização e na possibilidade de retreino com novos conjuntos de dados, então essa abordagem foi escolhida para ser explorada neste trabalho. Em [14], uma *vision transformers* é usada juntamente com um *distillation token*, obtendo um AUC de 0,978 no dataset DFDC. Zhao et. al [15] também empregam atenção ao utilizar uma rede com multi-atention para extrair características discriminativas de diferentes regiões faciais. Atenção também foi utilizada para realizar aumentação nos dados e ajudar no treino, que por conta da arquitetura escolhida não é trivial. Os experimentos foram conduzidos tanto em condições *intra-dataset* nos datasets FaceForensics++ High Quality e DFDC, quanto em experimentos *cross-dataset*, treinando-se no FaceForensics++ e testando no Celeb-DF. Os resultados obtidos *intra-dataset* têm acurácia de 97.6 e logloss de 0.1679 para FaceForensics++ e DFDC, respectivamente. As métricas *cross dataset* mostram uma boa transferibilidade do modelo com um AUC de 67,44, porém nota-se uma grande queda em relação as métricas *intra dataset*.

Em [16] os autores treinaram um classificador de uma única classe para suprir a deficiência destes modelos de detecção nos testes cross dataset. O método proposto possui várias características além de ser um classificador de uma única classe, como o uso de métodos de aprimoramento de filtros para melhorar os dados, a adoção de uma rede neural convolucional multi-canal aprimorada para extrair mapas de atenção e o treinamento em duas etapas para filtrar características falsas conhecidas e desconhecidas. O dataset ProGAN foi usado para treinamento e os métodos StyleGAN, StyleGAN2, BigGAN, DCGAN, DeepFake e VQ-VAE2.0 como os métodos de geração de faces falsas não vistas durante o treino. Em relação ao desempenho de detecção cross dataset, a acurácia do método proposto foi de 87.2% para StyleGAN, 85.1% para StyleGAN2, 82.7% para BigGAN, 93.1% para DCGAN, 89.7% para DeepFake e 80.9% para VQ-VAE2.0, respectivamente. Em comparação com outros métodos de detecção recentes, especialmente aqueles que não consideram a capacidade de generalização, a acurácia apresentou melhorias de 5% a 30% e no F1-Score de 105% a 205%.

Abordagens com foco nos dados em vez do modelo têm se mostrado populares e podem ser aplicadas a qualquer modelo existente. Na área de detecção de deepfake isso pode ser especialmente útil uma vez que os datasets existentes são, muitas vezes, altamente superamostrados, causando um fácil *overfit*. Isso ocorre pois esses conjuntos de dados são criados utilizando um pequeno conjunto

de rostos reais para gerar múltiplas amostras falsas. Em Wang et al. [17], a generalização do modelo melhorou significativamente apenas com *augmentation* simples nos dados, como desfoque e ruído aleatório. Já Das et. al [18] vão um pouco além ao fazer um apagamento aleatório de partes da face utilizando pontos fiduciais previamente detectados.

Wang et al. [19] utilizaram um mecanismo de atenção para detectar regiões mais sensíveis ao modelo e geraram mais dados apagando essas regiões, fazendo com que o modelo tivesse que aprender a "prestar atenção"em uma região maior da imagem. Como resultado, detectores baseados em CNN atingiram métricas de estado da arte sem alteração da arquitetura. Li et al. [20] também utilizaram aumentação de dados com atenção para fazer um crop nos dados e treinaram a rede em dois passos para buscar generalização. O primeiro passo utiliza uma função de perda binária para filtrar imagens geradas por ataques conhecidos, já o segundo passo utiliza uma função de perda one-class para conseguir distinguir ataques não vistos. As imagens reais utilizadas vieram do conjunto de dados Flickr-Faces-High-Quality (Flickr-Faces-HQ) [21], enquanto as imagens falsas foram geradas pelos próprios autores a partir deste conjunto de dados. Todas as técnicas utilizadas foram baseadas em GANs. A principal melhoria nas métricas obtidas foi em testes cross-domain, tendo uma melhora de 5% a 30% na acurácia e de 105% a 205% no F1-Score em comparação com modelos populares nesta tarefa. Em Zhang et al. [22], foram geradas características no espectro de frequência que simulam artefatos presentes em imagens falsas geradas por diversos modelos.

Com a introdução de modelos como o DALL-E 2, capazes de gerar imagens a partir de entradas textuais, além do impacto significativo na indústria artística, começou-se a explorar a utilidade desses modelos para aprimorar tarefas de visão computacional. Isso se deve ao fato de que um dos maiores desafios no treinamento de modelos confiáveis nessa área é obter um grande volume de dados de alta qualidade, o que poderia ser abordado utilizando modelos como o DALL-E 2 para gerar variações de imagens já anotadas. O estudo realizado por Sagers et al. [23] demonstrou que essa abordagem melhora a acurácia de classificadores de doenças de pele, especialmente em grupos sub representados. No entanto, é importante ressaltar que, conforme discutido em um estudo por outro autor em [24], também na área médica mas desta vez com dados de radiologia, o uso do DALL-E 2 para gerar e aumentar dados é promissor, mas pode exigir um ajuste fino (fine tuning) do modelo gerador de imagens antes de ser plenamente efetivo.

Tais modelos usam modelos de difusão para converter textos para imagens através de uma abordagem iterativa, onde começa-se com uma imagem inicial de ruído e, em cada etapa, aplica um processo de difusão para suavizar o ruído e gerar detalhes mais complexos. Ao longo das iterações, a imagem evolui gradualmente até alcançar uma representação realista. O StableDiffusion [25] está na mesma categoria de modelos e foi treinado em imagens 512x512 de um subconjunto do conjunto de dados LAION-5B [26], o maior conjunto de dados multimodal atualmente disponível. Com ele também é possível gerar variações de uma mesma imagem a partir de descrições textuais, além de possuir código aberto.

A criação adequada dos prompts para alimentar os modelos de difusão é de extrema importância, uma vez que as imagens resultantes dependem significativamente deles. No entanto, a tarefa de criação dos prompts pode ser altamente exigente em termos de recursos. Como resultado, pesquisas têm sido conduzidas para explorar abordagens mais eficientes nesse processo. No trabalho de Pavlichenko et al. [27], uma abordagem "human in the loop"é apresentada, empregando um algoritmo genético para a criação dos prompts. Por outro lado, Hao et al. [28] propõem melhorias ao gerar variações de maior qualidade dos prompts gerados por seres humanos, utilizando aprendizado por reforço, em que a função de recompensa incentiva o modelo de difusão a gerar imagens de maior qualidade, mantendo a intenção original do prompt.

Uma alternativa seria empregar Large Language Models (LLM) de geração de texto para criar esses prompts, esses modelos são treinados em grandes volumes de dados textuais e têm a capacidade

de responder perguntas, gerar diálogos naturais e até mesmo redigir textos criativos e coerentes. O ChatGPT, por exemplo, tem sido amplamente utilizado em diversas aplicações, como chatbots, assistentes virtuais, assistentes para escrita de código e até mesmo na criação de websites completos. Além disso, existem versões de código aberto, como o Llama (Large Language Model Meta AI) [29], desenvolvido pela Meta e treinado em textos das 20 línguas mais faladas, com foco nas línguas que utilizam os alfabetos latino e cirílico. Esse modelo de linguagem gera texto ao receber uma sequência de palavras como entrada e realiza previsões recursivas para a próxima palavra, seguindo a abordagem comum a outros modelos dessa categoria.

3 Materiais e Métodos

Esta seção apresenta os materiais e métodos empregados no presente estudo, que tem como objetivo avaliar o impacto do aumento de dados utilizando GANs e modelos image-to-image de difusão na capacidade de generalização de modelos de aprendizado profundo na detecção de deepfakes para dados de domínios não vistos.

Inicialmente, descrevemos o ambiente de treinamento utilizado, destacando as configurações e recursos computacionais disponíveis. Em seguida, discutimos o processo de seleção dos conjuntos de dados utilizados nos experimentos, levando em consideração critérios como qualidade e volume dos dados.

Posteriormente, detalhamos o pré-processamento realizado nos dados, abrangendo etapas como extração dos quadros e detecção facial. Em seguida, apresentamos as diferentes técnicas de aumento de dados empregadas, com destaque para o uso de GANs e modelos image-to-image de difusão, ressaltando as razões para a escolha dessas abordagens e os parâmetros utilizados.

Por fim, apresentamos o modelo que será testado, abordando sua arquitetura e características específicas, e descrevemos o protocolo de treinamento adotado, incluindo informações sobre o número de épocas, taxa de aprendizado e algoritmo de otimização utilizado.

3.1 Ambiente dos experimentos

O servidor utilizado neste estudo foi uma máquina com um processador multi-core de 64 núcleos e memória RAM de 128GB. Além disso, o sistema é equipado com uma GPU NVIDIA Titan RTX, possuindo 24GB de VRAM e projetada especificamente para acelerar o processamento paralelo exigido pelo aprendizado profundo.

Apesar da capacidade considerável do sistema, é importante mencionar que alguns recursos de hardware ainda apresentaram limitações durante a realização dos experimentos. O espaço em disco disponível para uso foi de 200GB, o que impôs restrições ao armazenamento de conjuntos de dados e modelos de grande escala, também, apesar da GPU possuir 24GB de VRAM, isso ainda não foi o suficiente para rodar o modelo de geração textual. No entanto, medidas foram tomadas para gerenciar o espaço em disco de forma eficiente, como a remoção de dados não essenciais e o uso de técnicas de compactação quando apropriado, como quantização dos modelos.

Em termos de software, os experimentos foram conduzidos todos em Python e utilizando frameworks de aprendizado de máquina de código aberto, como PyTorch, que fornecem interfaces eficientes para aproveitar o poder computacional do hardware disponível. Além disso, bibliotecas especializadas foram utilizadas para o pré-processamento de dados, aumento de dados e avaliação de desempenho dos modelos. Mais detalhes podem ser encontrados nas seções abaixo.

3.2 Datasets

Dado que o objetivo principal deste trabalho consiste em investigar a melhoria do desempenho do modelo em relação a dados de contexto não vistos, tornou-se crucial a obtenção de conjuntos de dados provenientes de diversas fontes. Nesse sentido, foram selecionados três conjuntos de dados públicos que apresentam diferentes técnicas de geração de vídeos falsos e também diferem em relação à origem dos vídeos originais.

Outros fatores também foram levados em consideração na escolha dos bancos de dados, como facilidade de acesso aos dados, tamanho em gigabytes (GB), quantidade de dados disponíveis, equilíbrio geral em termos de gênero, cor, idade e proporção de vídeos reais e falsos. A disponibilidade de acesso direto foi um critério relevante devido à necessidade de evitar atrasos decorrentes da obtenção de aprovações prévias para utilização de determinados conjuntos de dados. Além disso, o tamanho também foi um grande fator limitante devido ao espaço em disco disponível, uma vez que todos os conjuntos de dados consistem em milhares de vídeos, alguns dos quais ultrapassam 1TB de dados.

Os três conjuntos de dados selecionados foram: DFDC [30], FaceForensics++[31] e CelebDF V2 [32]. Tais conjuntos de dados diferem em relação às técnicas de geração de deepfake e à fonte dos vídeos originais, enquanto o dataset DFDC apresenta vídeos originais gravados com atores pagos especificamente para a criação do conjunto de dados, os conjuntos FaceForensics++ e CelebDF v2 contêm vídeos obtidos do YouTube. Além disso, há variações nas técnicas de geração de deepfake utilizadas em cada conjunto. No DFDC, a maioria dos vídeos foi gerada utilizando a técnica DFAE, enquanto no FaceForensics++ são utilizadas uma combinação de técnicas com aprendizado (DFAE e FaceShifter) e sem aprendizado (FaceSwap). Por sua vez, no CelebDF v2, foi empregado um DFAE aprimorado. O DFDC e o CelebDF contém apenas vídeos de troca de identidade, já o FaceForensics contém vídeo de troca de identidade (DFAE, FaceSwap e FaceShifter) e também troca de expressão (Face2Face e NeuralTextures), porém, como o foco deste trabalho é detectar trocas de identidade, então não foram utilizados os vídeos de troca de expressão deste conjunto.

Essa diversidade nos conjuntos de dados contribui para uma análise mais abrangente das capacidades de generalização dos modelos de detecção de deepfake, permitindo a avaliação de seu desempenho em relação a diferentes tipos de deepfakes e cenários de geração de vídeos falsos. No entanto, é importante ressaltar que os conjuntos de dados selecionados não garantem um equilíbrio adequado em termos de gênero, etnia e idade. Embora tenha sido considerada a existência de conjuntos de dados com essas características, sua utilização foi inviável devido ao tamanho excessivo. No entanto, esse aspecto específico não é o foco principal deste trabalho e, portanto, foi tratado como um detalhe secundário que pode ser negligenciado.

3.3 Pré-processamento

O pré-processamento dos vídeos desempenha um papel fundamental neste estudo, especialmente porque os modelos testados são baseados em imagens. Os seguintes passos foram realizados:

3.3.1 Extração das imagens

Inicialmente, foi necessário converter os vídeos de seus formatos originais (.mp4, .mov, .avi, etc.) para uma sequência de imagens, uma vez que os modelos são baseados em imagens. Esse processo foi realizado utilizando a biblioteca OpenCV [33], preservando-se a qualidade, dimensão e taxa de quadros original dos vídeos.

3.3.2 Amostragem dos frames

Considerando que as imagens são extraídas de vídeos, muitas delas são semelhantes entre si, o que pode resultar em viés durante o treinamento do modelo. Portanto, foi realizada uma amostragem dos frames, selecionando apenas uma parte deles. Após uma análise geral do conteúdo dos vídeos, constatou-se que o número 20 oferece um bom equilíbrio entre a redução da redundância e a preservação da variedade de informações nas imagens.

3.3.3 Recorte da face

Em tarefas de detecção de fraudes faciais, é comum e crucial realizar a detecção e o recorte da região facial antes de fornecer as imagens ao modelo, pois faz com que o modelo foque no que é relevante, facilita a normalização da entrada e também reduz a complexidade do problema ao reduzir o tamanho das imagens. Para esse fim, foi utilizada a ferramenta FaceMesh do MediaPipe [34], que é um modelo de aprendizado de máquina desenvolvido pela Google para detecção e rastreamento de pontos faciais em tempo real. O FaceMesh é projetado para identificar e rastrear 468 pontos-chave do rosto humano, incluindo olhos, sobrancelhas, nariz, boca e contorno facial. Essa ferramenta foi escolhida devido à sua rapidez, precisão e facilidade de uso.

Para garantir uma maior qualidade das faces utilizadas, na detecção facial foi escolhido um limiar pra confiança da detecção de 0.5. Assim, imagens com confiança abaixo desse limiar foram descartadas. A Figura 2 traz exemplos de imagens antes e após o recorte facial. Nota-se a grande quantidade de informação que é removida ao ignorar o fundo das imagens.

(a) CelebDF - Face real sem recorte

(b) CelebDF - Face real após recorte

(c) FF++ - Face falsa sem recorte

(d) FF++ - Face false após recorte

Figura 2: Exemplos de imagens reais e falsas antes e depois do recorte

3.3.4 Split

Dado que o objetivo deste trabalho consiste em avaliar se a aplicação de técnicas de aumento de dados utilizando modelos de aprendizado de máquina melhora a capacidade de generalização de modelos de detecção de deepfake para ataques não vistos, é fundamental estabelecer um protocolo de treinamento que permita a obtenção de métricas intra e cross dataset. As métricas intra dataset são obtidas a partir de dados de teste que contêm imagens de contextos observados durante o treinamento, enquanto as métricas cross dataset são obtidas a partir de dados em que o contexto não foi visto durante o treinamento.

Logo, inicialmente foi necessário escolher dois conjuntos de dados para treinamento, validação e teste intra dataset, e um conjunto para teste cross dataset. Optou-se por utilizar o conjunto de teste oficial do DFDC para os testes cross dataset, devido à sua maior abrangência e representatividade em termos de ambiente, iluminação e diversidade de indivíduos. Os conjuntos FaceForensics++ e CelebDF V2 foram selecionados para o treinamento, validação e testes intra dataset, pois apresentam uma variedade de técnicas de geração de deepfake, contribuindo para a construção de um modelo mais robusto e generalizável. É importante mencionar que foram excluídos do conjunto de teste oficial do DFDC os vídeos não originais, que foram editados para aumentar o volume de dados, a fim de evitar qualquer influência indesejada nos resultados.

Após a seleção dos conjuntos de dados, foram realizadas as divisões para treinamento, validação e teste pros conjuntos CelebDF e FaceForensics++. No conjunto de dados CelebDF, a separação prévia dos dados de teste já estava disponível, e a divisão restante foi feita com uma proporção de 90% para treinamento e 10% para validação. Os dados reais desse conjunto são divididos entre os conjuntos "youtube"e "celeb", portanto a divisão entre treino a validação foi feita buscando-se manter a proporção dessa divisão em ambos os conjuntos, para garantir um equilíbrio adequado.

Por outro lado, o conjunto de dados FaceForensics++ não apresenta uma separação prévia para o conjunto de teste. Portanto, foi realizada uma divisão utilizando uma proporção de 80% para treinamento, 10% para validação e 10% para teste. Nesse caso, como a separação do conjunto de teste foi feita manualmente, foi possível tomar o cuidado de realizar a separação por indivíduo, evitando a presença de um mesmo indivíduo em conjuntos diferentes. Isso é importante para evitar que o modelo aprenda a identificar faces em vez de distinguir entre vídeos reais e falsos, nota-se que para os outros dois datasets isso não é garantido. Inicialmente, foram identificados todos os indivíduos únicos nos dados reais, realizando a separação deles nos conjuntos de treinamento, validação e teste. Em seguida, utilizaram-se esses mesmos indivíduos como referência para separar os dados falsos, levando em consideração a face "source"dos dados falsos, que representa a face final exibida nos vídeos. Além disso, como os vídeos falsos são separados de acordo com três diferentes técnicas (Deepfakes, FaceShifter e FaceSwap), o split foi feito individualmente em cada uma dessas separações e também nos dados reais, garantindo uma distribuição equilibrada de cada tipo nos conjuntos de treinamento, validação e teste.

É importante ressaltar que a separação dos conjuntos de treinamento, validação e teste não considerou características como gênero, idade e etnia, uma vez que essas informações não estavam anotadas nos datasets utilizados. Idealmente, a consideração desses aspectos seria desejável para evitar qualquer viés indesejado no modelo treinado.

A Tabela 1 resume cada dataset utilizado após extração das imagens, amostragem dos frames, recorte da face e separação dos conjuntos, note que o número de imagens final é reduzido devido à detecções faciais de baixa confiança, principalmente para o dataset DFDC, cuja qualidade é menor.

Dataset	Split	Classe	Nº videos	Nº final de imagens
CelebDF	Treino	Falso	4769	29385
		Real	641	9122
	Validação	Falso	530	8081
		Real	71	1040
	Teste	Falso	370	5158
		Real	178	2508
FaceForensics++	Treino	Falso	2400	27112
		Real	800	8239
	Validação	Falso	300	3445
		Real	100	1090
	Teste	Falso	300	3365
		Real	100	1070
DFDC	Teste	Falso	536	2658
		Real	504	1785

Tabela 1: Quantidade final de vídeos e imagens por dataset e split.

3.4 Aumento dos dados

Nesta seção são descritas as três formas de aumento de dados testadas neste trabalho: aumento simples, com modelo de difusão e com GANs. Adicionalmente, as imagens de treino em todas as abordagens (exceto o treino apenas com os dados originais) foram submetidas a uma técnica denominada *random erase* com uma probabilidade de 0.3. Essa abordagem consiste em apagar aleatoriamente uma região retangular nas imagens e substituí-la por valores aleatórios. Essa estratégia tem como objetivo reduzir o risco de overfitting e melhorar a robustez do modelo diante de oclusões [35].

3.4.1 Aumento simples

O aumento de dados com técnicas simples foi feito para fins de comparação, já que eles são a forma mais de comum de realizar estar tarefa. Neste trabalho isso foi feito usando o processo embutido que há no *timm*, mais especificamente, foi utilizado o RandAugment, que em cada aumento aplica N transformações com magnitude M a partir de uma lista de transformações disponíveis. Nos experimentos deste trabalho, para cada imagem original foram geradas duas novas imagens e, por se tratar de uma tarefa onde os artefatos de detecção podem ser sutis, foi usada apenas uma transformação para cada aumento, uma magnitude de 3 cada uma e um desvio padrão de 0.5 nessa magnitude.

Há diversas transformações disponíveis no *timm*, como mudança equalização de histograma, mudança aleatória na nitidez, rotação, etc. Para escolher as que seriam aplicadas tomou-se o cuidado de não utilizar transformações que pudessem ter efeitos semelhantes à artefatos presentes em imagens falsas geradas, como mudanças na nitidez da imagem. Sendo assim, a lista de transformações usadas foi:

- Identity
- AutoContrast
- Equalize
- Rotate
- Solarize
- Color
- Posterize
- Contrast
- Brightness
- ShearX
- ShearY
- TranslateX
- TranslateY

Um exemplo de um conjunto de imagens aumentadas é apresentado na Figura 3.

Figura 3: Exemplo de imagens aumentadas através de técnicas simples juntamente com *random erase* com probabilidade de 0.3.

3.4.2 Aumento com modelo de difusão image to image

Neste tipo de aumento de dados dois modelos multimodais foram empregados: um modelo gerador de texto e um modelo de difusão *img2img*. O modelo *img2img* recebe uma imagem como entrada juntamente com um prompt contendo a variação desejada. Para gerar tais comandos, o modelo gerador de texto foi utilizado, uma vez que realizar essa tarefa manualmente seria dispendioso e também devido ao conhecimento que a vasta quantidade de informações que esse modelo textual possui pode agregar.

A escolha desses modelos levou em consideração o fato de possuírem código aberto, uma vez que utilizar serviços conhecidos, como ChatGPT e DALL·E, implicaria em altos custos. Além disso, dada a disponibilidade de um hardware com GPU, executar esses modelos localmente é viável.

Criação de prompts para alimentar o modelo text to text:

Para a geração de texto, utilizou-se o Llama [29], modelo que possui código aberto e disponível em diferentes tamanhos (7B, 13B, 33B e 65B de parâmetros). A princípio rodá-lo mostrou-se inviável por conta da GPU disponível não ter capacidade, porém foi contornar esta situação ao quantizar o modelo, usando-se assim 16GB de VRAM na GPU e 8GB de memória RAM pra versão 7B, que acabou sendo a única viável.

Foi desenvolvido um prompt apropriado para alimentá-lo, o qual é responsável por gerar instruções de variação de imagem para o modelo *image-to-image*. Esse prompts consistem em frases ou instruções que indicam as alterações desejadas nas imagens de treinamento, tais como mudanças de cor, contraste, textura e outros aspectos. O modelo "text to text"utilizado opera ao completar o prompt de entrada, recebendo, portanto, texto contendo instruções para gerar frases de variação, juntamente com exemplos de como isso seria feito.

O prompt de entrada utilizado em todas as iterações foi o seguinte:

Entrada:
List of face image variations while keeping the overall structure of the original image intact, no repetitions and no long sentences:
Different lighting conditions
Altered facial expressions
Different hairstyle
Variations in makeup styles
Variations in ethnicity
Background is a blue sky
Angry
Crying
Eyes are closed
Smiling
Background is fire
Blue hairstyle

Um exemplo de retorno obtido é:

List of face image variations while keeping the overall structure of the original image intact, no repetitions and no long sentences:
Different lighting conditions
Altered facial expressions
Different hairstyle
Variations in makeup styles
Variations in ethnicity
Background is a blue sky
Angry
Crying
Eyes are closed
Smiling
Background is fire
Blue hairstyle
Background i s a blue house
Smiling

Variations in clothing style
Different hair length
Different hair texture

Foram realizadas 100 iterações com o mesmo prompt de entrada, gerando 150 novos tokens a cada vez, com o parâmetro **temperature** configurado como 0.35. Um valor de **temperature** baixo resultava em muitas frases repetidas, enquanto um valor alto gerava textos sem relação com o objetivo da tarefa, como trechos de código e letras de músicas. O valor final foi encontrado experimentalmente.

Além disso, a fim de eliminar ruídos indesejados, como frases já listadas ou frases contendo palavras duplicadas (por exemplo, "Face is happy and smiling and happy and smiling", identificadas durante os primeiros testes possivelmente devido ao uso do modelo com menor número de parâmetros), foram realizadas duas etapas de pós-processamento: 1) aplicação de um filtro utilizando o tokenizador do scikit-learn [36], em conjunto com a lista de palavras irrelevantes (stop words) do nltk [37], para eliminar frases que continham tokens repetidos após o processo de tokenização e remoção das palavras irrelevantes; e 2) seleção apenas de frases que ainda não haviam sido selecionadas.

Ao fim deste processo obteve-se um total de 359 prompts de entrada para o modelo *image to image*.

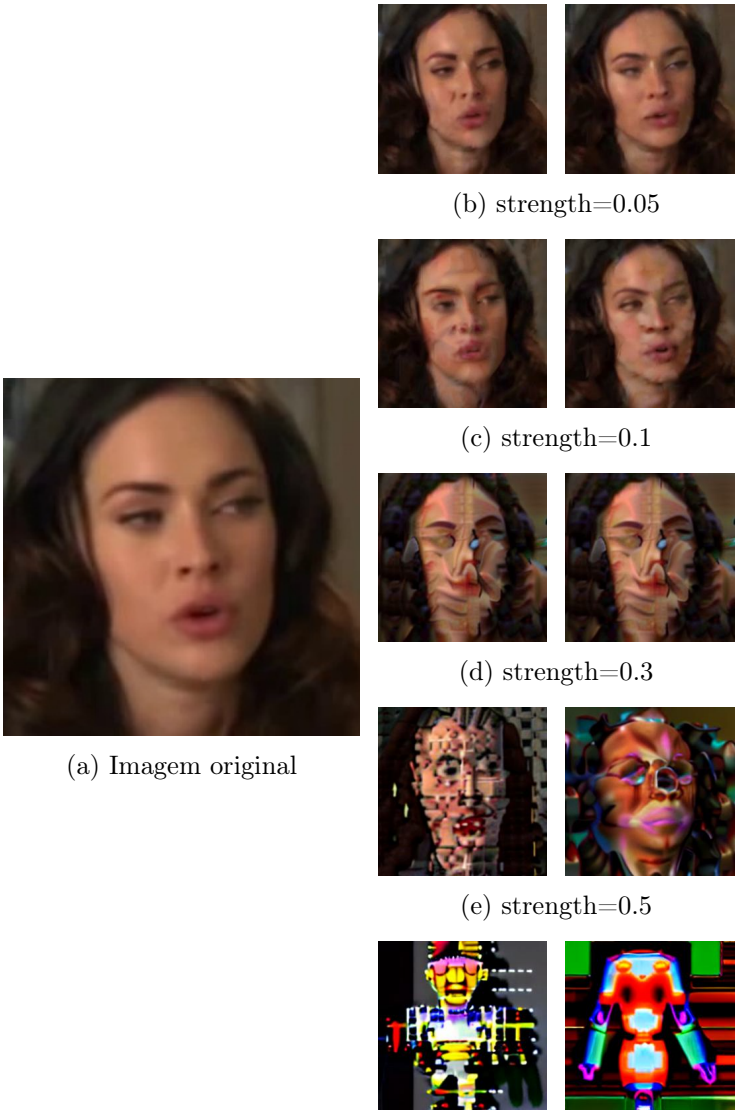
Utilização dos prompts gerados para gerar variações das imagens de treino:

O modelo selecionado para a geração de imagens foi o Stable Diffusion e a execução deste modelo localmente demandou 18GB de VRAM na GPU e 8GB de memória RAM. Neste passo, os prompts de variação de imagem gerados anteriormente são utilizados para realizar modificações nas imagens de treinamento. Utilizando o modelo *image-to-image*, esses prompts são aplicados às imagens, resultando na geração de imagens aumentadas que incorporam as variações desejadas. Esse processo permite aumentar a diversidade dos dados de treinamento.

O modelo original do Stable Diffusion foi treinado no dataset LAION-5B [26], que contém bilhões de pares de imagem-texto de diversos domínios. No entanto, os pesos utilizados neste trabalho são os "stable-diffusion-v1-5", que foram treinados por meio de um ajuste fino (finetuning) do modelo original no dataset LAION-Aesthetics V2. Esse dataset consiste em um subconjunto do LAION-5B, porém com imagens de melhor qualidade.

Como a quantidade de prompts é significativamente menor do que a quantidade total de imagens para treinamento, a utilização desses prompts foi realizada de maneira circular. Para cada imagem, foram utilizados dois prompts, gerando assim duas novas imagens. Quando todos os prompts foram utilizados, a lista foi reiniciada. Essa quantidade foi escolhida considerando que o aumento de dados com RandAugment também gerou duas novas imagens para cada imagem original. Além disso, a geração de um grande número de novas imagens seria inviável devido às limitações de espaço em disco mencionadas anteriormente.

Um parâmetro importante nessa etapa é a *strength* do modelo *image-to-image*. Assim como a *temperature* do modelo anterior, esse parâmetro varia de 0 a 1, onde valores mais próximos de 1 resultam em saídas mais aleatórias e distantes da imagem de entrada. Foi realizada uma busca para encontrar o melhor valor para esse parâmetro, visualizando imagens geradas para diferentes valores de *strength* a partir de uma mesma imagem fonte e dois prompts diferentes. Alguns dos resultados obtidos podem ser observados na Figura 4.



(a) Imagem original

(b) strength=0.05

(c) strength=0.1

(d) strength=0.3

(e) strength=0.5

(f) strength=0.8

Figura 4: Imagem original e suas variações geradas pelo StableDiffusion com diferentes strengths para prompt 1 (imagem à esquerda) e prompt 2 (imagem à direita). Prompt 1: "Different hair length"; Prompt 2: "Different hair texture".

Observa-se que as alterações feitas pelo modelo são muito abstratas e aleatórias, o que não era o esperado, uma vez que se acreditava que a diversidade dos dados de treinamento seria suficiente para compreender as instruções de alteração facial desejadas. Portanto, foi necessário utilizar um valor de strength bastante baixo, de 0.05, para garantir que as características das imagens fossem preservadas.

3.4.3 Aumento com StyleGANs

Este tipo de aumento consiste na utilização de StyleGANs para gerar novas faces, com o objetivo de enriquecer o conjunto de treinamento por meio da inclusão de imagens sintéticas representando variações não presentes nos dados originais. Para essa finalidade, foi empregado a SemanticStyleGANs

[38], uma variante da StyleGANs em que o gerador é treinado para modelar separadamente as diferentes partes semânticas das faces, permitindo a síntese das faces por meio da composição de vetores específicos para cada parte. Esse método resulta em resultados surpreendentes. Adicionalmente, o código para treinamento e geração das faces está disponível publicamente.

Inicialmente, o modelo é retreinado utilizando os dados do contexto específico em que se busca melhorar a generalização, neste caso os conjuntos de faces falsas dos datasets CelebDF e FaceForensics++. Essa etapa visa adaptar o modelo gerativo às características e padrões específicos dos dados, assegurando uma geração mais coerente e relevante dentro do contexto considerado . Optou-se por retreinar apenas no conjunto de imagens falsas, uma vez que o objetivo é aumentar especificamente essa parte dos dados. O aumento do conjunto de imagens reais poderia introduzir inconsistências na tarefa, uma vez que as novas faces geradas seriam falsas.

Foram realizados dois tipos de retreinamento, com e sem a ativação do parâmetro *freeze locals*. Esse parâmetro congela os geradores locais, preservando a separação espacial do treinamento original do modelo. No entanto, não é recomendado para ajuste em datasets com uma grande diferença em relação às faces reais. Optou-se por realizar ambos os tipos de treinamento, uma vez que é difícil determinar se as faces falsas estão suficientemente distantes das faces reais treinadas originalmente. Portanto, a melhor opção é observar isso experimentalmente. A Figura 5 apresenta uma comparação das imagens obtidas com esse parâmetro ativado e desativado, sendo observado que o resultado obtido com o parâmetro desativado é mais fiel às imagens desejadas. Por essa razão, as imagens geradas pelo modelo treinado dessa forma foram escolhidas para uso.

Após o retreinamento do modelo StyleGAN, foram geradas 27.901 novas imagens aleatórias. Esse número foi escolhido para equilibrar a quantidade de imagens falsas nesse "novo"conjunto com os conjuntos existentes. A utilização do StyleGAN como técnica de aumento de dados tem o objetivo de enriquecer a diversidade e representatividade do conjunto de treinamento. Espera-se que isso permita que o modelo de detecção de deepfake aprenda a reconhecer e distinguir de maneira mais eficaz as características distintivas entre faces reais e deepfakes em domínios não vistos, ou seja, no dataset DFDC.

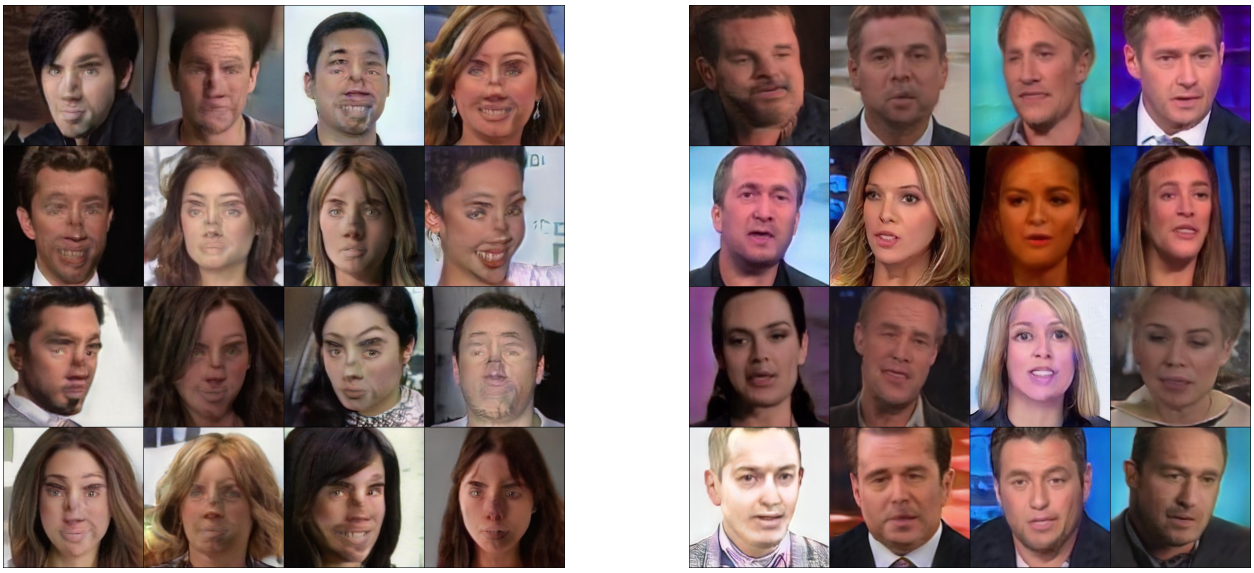


Figura 5: Comparação entre imagens geradas após retreino da SemanticStyleGANs no conjunto de faces falsas do CelebDF e FaceForensics++ para o parâmetro *freeze local* como True (imagem à esquerda) e como False (imagem à direita)

3.5 Escolha dos modelos e protocolo de treinamento

3.5.1 Modelos

Os modelos Vision Transformers (ViT) têm demonstrado um grande potencial em tarefas de classificação de imagens, apresentando resultados surpreendentes [39]. Esses modelos são baseados na arquitetura dos transformers, semelhante ao BERT, e são pré-treinados e ajustados em uma extensa coleção de imagens supervisionadas, como o ImageNet-1k, com resolução de 224x224 pixels. As imagens são apresentadas ao modelo como uma sequência de patches de tamanho fixo (resolução de 16x16 pixels), que são incorporados linearmente. Adicionalmente, um token [CLS] é adicionado no início da sequência para uso em tarefas de classificação. Além disso, são adicionados embeddings de posição absoluta antes de alimentar a sequência nas camadas do codificador. As vantagens desses modelos são diversas, destacando-se seu design simples, que permite o processamento de diferentes modalidades de dados (imagem, vídeo, texto, áudio, etc.), além de demonstrar escalabilidade em relação ao tamanho da rede e ao número de dados.

Dessa forma, optou-se por realizar experimentos com uma variante do ViT, chamada DeiT (Data-efficient Image Transformer) [40], a partir de seus pesos pré-treinados. Essa escolha foi motivada pela capacidade dessa rede de obter bons resultados com uma quantidade menor de dados, devido à sua arquitetura e ao treinamento baseados em uma estratégia teacher-student, juntamente com um *distillation token*. Mais especificamente, utilizou-se a versão *deit-base-patch16-224*.

Para testar o modelo DeiT e realizar o reajuste (a partir dos pesos pré-treinados), utilizou-se o PyTorch Image Models (timm) [41], uma biblioteca em Python que oferece uma variedade de modelos de aprendizado profundo para tarefas de visão computacional utilizando o framework PyTorch. Essa biblioteca foi escolhida devido à sua facilidade de uso e à possibilidade de experimentação com diversas arquiteturas de redes neurais pré-treinadas, como EfficientNet, ResNet, ViT, DeiT e outras. O timm fornece uma implementação eficiente e flexível desses modelos, além de recursos adicionais, como transformações de dados, treinamento e avaliação de modelos, otimização e aumento de dados, todos utilizados neste projeto.

3.5.2 Protocolo de treinamento

Para cada treinamento feito, realizou-se um teste intra dataset e um teste cross dataset, onde todos esses dados não sofreram alteração nenhuma. Sendo assim, foram realizados seis treinamentos distintos e para cada um atribuiu-se um número para identificação nas discussões seguintes:

1. Apenas dados originais, sem aumento
2. Aumento de dados simples
3. Aumento de dados com modelo de difusão
4. Aumento de dados com modelo de difusão + aumento simples
5. Aumento de dados com GANs
6. Aumento de dados com GANs + aumento simples

Essa estratégia permite uma comparação clara do efeito do aumento de dados na capacidade de generalização do modelo. O conjunto de validação manteve-se o mesmo em todos os treinamentos, buscando-se uma maior semelhança entre ele e o conjunto de teste intra dataset. O modelo escolhido

para teste foi o modelo com menor loss de validação.

O tamanho do batch foi definido como 64, permitindo um equilíbrio entre eficiência computacional e estabilidade no processo de treinamento. O treinamento foi realizado por um mínimo de 25 épocas, sendo interrompido caso a perda de validação deixasse de diminuir por 5 épocas consecutivas, garantindo assim a convergência do modelo. O valor inicial da taxa de aprendizado foi estabelecido em 0,005 e foi utilizado um scheduler cosseno, que reduz gradualmente a taxa ao longo do tempo. A função de perda utilizada foi a binary cross entropy loss, adequada para problemas de classificação binária.

3.5.3 Métricas

A taxa de ocorrência de vídeos fraudulentos com deepfake no mundo real é consideravelmente baixa em comparação com vídeos genuínos. Nesse contexto, ao abordarmos o problema como uma classificação binária, em que a classe fraude é considerada positiva, é crucial que o modelo apresente uma baixa taxa de falsos positivos, pois mesmo uma taxa baixa pode resultar em uma classificação equivocada de muitos vídeos. Além disso, a precisão não é sensível ao desbalanceamento de classes presente nos conjuntos de teste, logo ela emerge como a métrica mais relevante neste trabalho, no entanto, também serão avaliadas a acurácia e o recall, pois são métricas úteis para compreender o comportamento do modelo.

4 Resultados

Neste capítulo, apresentamos e discutimos os resultados obtidos com a aplicação da metodologia descrita. A Tabela 2 resume as métricas obtidas para cada treino feito, onde as melhores métricas de cada contexto estão em destaque.

Treino	**Melhor época**	**Val**			**Intra Dataset**			**Cross Dataset**		
		Acc	**Prec**	**Recall**	**Acc**	**Prec**	**Recall**	**Acc**	**Prec**	Recall
1	15	0.9722	0.9834	0.9837	0.9637	0.9681	0.9807	**0.7668**	0.9154	0.6723
2	16	**0.9870**	**0.9896**	**0.9951**	**0.9751**	**0.974**	**0.991**	0.758	0.8832	**0.6862**
3	22	0.934	0.949	0.9742	0.8911	0.89	0.9646	0.7346	0.8948	0.6305
4	7	0.9624	0.9756	0.9799	0.9422	0.9444	0.9753	0.7539	**0.9448**	0.6252
5	20	0,9709	0,9802	0,9854	0,956	0,9568	0,9818	0,7539	0,8863	0,6749
6	31	0,9836	0,9876	0,9931	0,9743	0,9739	0,9901	0,7582	0,9078	0,6632

Tabela 2: Resultados dos treinamentos feitos, a numeração dos treino segue a descrita na seção 3.5.2

Observou-se que os treinamentos que empregaram apenas o aumento simples dos dados apresentaram os melhores resultados em termos de métricas de validação e intra dataset. No entanto, nos testes cross dataset, verificou-se que a maior acurácia foi alcançada nos treinamentos que não utilizaram nenhum tipo de aumento, ao passo que o melhor recall foi obtido nos treinamentos com aumento simples. Apesar de tais resultados contrariarem as expectativas iniciais, destaca-se que o treinamento que empregou o aumento utilizando o modelo de difusão obteve a maior precisão nos testes cross dataset, demonstrando uma diferença significativa em relação aos demais modelos. Esse achado é surpreendente, considerando-se que as amostras de aumento utilizadas nesse caso se apresentavam incoerentes e aleatórias, o que pode explicar o baixo recall obtido por esse mesmo modelo.

Além disso, outra expectativa não atendida diz respeito às baixas métricas obtidas pelos modelos treinados com o uso de GANs para o aumento de dados, já que as imagens geradas por esses modelos apresentavam semelhanças visuais com as imagens falsas presentes nos conjuntos de dados CelebDF e FaceForensics++. Uma possibilidade para melhorar os resultados nessa abordagem seria fazer o ajuste do SemanticStyleGANs por mais épocas e em mais dados de faces falsas. Outro experimento válido seria gerar imagens falsas a partir de um treinamento realizado nas imagens reais, ou seja, criar imagens falsas "genuínas"em vez de apenas aumentar as imagens falsas já existentes, o que seria semelhante a adicionar um novo conjunto de dados ao treinamento. De qualquer forma, isso levanta a questão de se a alta precisão obtida no treinamento 4 foi realmente resultado dos dados ou de algum efeito aleatório.

A fim de investigar tal aleatoriedade, seria recomendável adotar um protocolo de treinamento que varie as combinações de conjuntos de treino, validação e teste. Para tal, seriam realizados três conjuntos de treinamento distintos: (1) Utilização de CelebDF e FaceForensics++ como conjuntos de treinamento/validação, e DFDC como conjunto de teste cross dataset; (2) Utilização de CelebDF e DFDC como conjuntos de treinamento/validação, e FaceForensics++ como conjunto de teste cross dataset; (3) Utilização de DFDC e FaceForensics++ como conjuntos de treinamento/validação, e CelebDF como conjunto de teste cross dataset. Esse procedimento permitiria reduzir a influência da seleção dos conjuntos de dados no desempenho dos modelos, fornecendo uma visão mais precisa do impacto real das técnicas empregadas. No entanto, devido a limitações de tempo para a realização do trabalho e restrições temporais para utilizar o hardware utilizado nos treinamentos, essa estratégia não pôde ser adotada.

Caso essa limitação não existisse também seria interessante explorar mais o abordagem do treino 4 dada a alta precisão cross dataset obtida nele, um experimento interessante seria submeter o modelo *image to image* a um ajuste fino no conjunto de dados do contexto da tarefa, uma vez que ele foi treinado originalmente em um dataset com imagens de características variadas, como obras de arte, imagens digitais, entre outras. Todavia, apesar desse ajuste poder ser útil nesse contexto, isso exigiria a tarefa custosa de anotar tais imagens com descrições textuais delas. Também, o uso de um modelo pré treinado com mais parâmetros, tanto para o mapeamento de imagem para imagem quanto para o mapeamento de texto para texto, poderia ter gerado resultados mais promissores, embora isso exigisse um hardware mais poderoso.

Outro ponto que pode ter afetado negativamente os resultados é a capacidade do modelo DeiT de detectar ataques não previamente observados, a arquitetura e a função de perda utilizadas no treinamento podem não ter sido as mais adequadas uma vez que são projetadas para classificação multi classe, sendo a classificação binária abordada neste projeto apenas uma instância de todas as possibilidades existentes. Por exemplo, seria interessante explorar a viabilidade de um modelo baseado em classificação "one class"projetado especificamente para distinguir instâncias "normais"ou "conhecidas"de instâncias "não observadas", empregando uma função de perda construída de maneira específica para tal propósito e viabilizando o uso desbalanceado de dados de faces reais, as quais são mais abundantes do que as faces falsas. Um exemplo dessa abordagem no contexto de detecção de deepfake pode ser encontrado em [16].

Ademais, outras estratégias poderiam ter sido consideradas com o intuito de aprimorar os resultados obtidos, como a realização de um particionamento dos conjuntos de dados CelebDF e DFDC por indivíduo, a condução de uma busca mais extensa pelos melhores hiperparâmetros tanto para o modelo de classificação quanto para os modelos geradores de dados e a ampliação do conjunto de treinamento do classificador mediante o aproveitamento dos outros conjuntos de dados disponíveis para essa tarefa.

5 Conclusão

Neste trabalho, avaliamos como o aumento de dados afeta a capacidade de um modelo em detectar imagens de faces falsas cujo contexto não foi visto durante o treinamento. Para isso, investigamos cenários intra dataset, nos quais o domínio dos dados de teste é o mesmo do treinamento, e cenários cross dataset, nos quais os dados de teste são provenientes de conjuntos de dados não utilizados durante o treinamento.

Os resultados obtidos corroboram a dificuldade na obtenção de métricas satisfatórias em contextos não vistos durante o treinamento na tarefa de detecção de deepfakes de troca de identidade em imagens faciais. Eles também indicam que o aumento de dados utilizando modelos *image-to-image* e GANs pode ser uma abordagem promissora para melhorar tais métricas. No entanto, tal abordagem envolve uma série de etapas que têm um impacto significativo nas métricas finais, como a seleção criteriosa dos conjuntos de dados adequados, o pré-processamento dos dados, a escolha do modelo classificador, a seleção dos modelos geradores de dados, a definição de um protocolo de treinamento eficaz, a busca por hiperparâmetros adequados, entre outros. Portanto, é necessário realizar mais experimentos a fim de explorar completamente as possibilidades e limitações destas estratégias.

Além disso, é relevante destacar a dificuldade de lidar com uma grande quantidade de dados e modelos que possuem uma quantidade massiva de parâmetros, muitas vezes da ordem de bilhões. A coleta, organização e processamento desses dados demandam recursos consideráveis em termos de tempo, esforço e capacidade de armazenamento. Além disso, o hardware utilizado pode ser um limitador, uma vez que modelos mais complexos requerem poder computacional adicional e treinamentos mais extensos. É fundamental reconhecer essas limitações e buscar soluções viáveis para lidar com esses desafios técnicos e computacionais.

Em suma, diante da ameaça significativa que os deepfakes representam para a sociedade, a pesquisa e o desenvolvimento contínuo de modelos de detecção de alta performance são de extrema importância. O estudo do efeito do aumento de dados com modelos *image-to-image* e GANs na capacidade de generalização dos modelos detectores de deepfakes para ataques não vistos oferece insights valiosos, reforça-se então a necessidade de realizar mais experimentos e investir em recursos computacionais para explorar todo o potencial destas abordagens. Através desses esforços, poderemos aprimorar a segurança e a confiança nas mídias digitais, protegendo a integridade da informação e mitigando os riscos associados aos deepfakes.

Referências

[1] M. Westerlund, "The emergence of deepfake technology: A review," *Technology innovation management review*, vol. 9, no. 11, 2019.

[2] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with clip latents," *arXiv preprint arXiv:2204.06125*, 2022.

[3] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401–4410, 2019.

[4] Z. Akhtar, "Deepfakes generation and detection: A short survey," *Journal of Imaging*, vol. 9, no. 1, 2023.

[5] I. Korshunova, W. Shi, J. Dambre, and L. Theis, "Fast face-swap using convolutional neural networks," in *Proceedings of the IEEE international conference on computer vision*, pp. 3677–3685, 2017.

[6] Y. Nirkin, I. Masi, A. T. Tuan, T. Hassner, and G. Medioni, "On face segmentation, face swapping, and face perception," in *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pp. 98–105, IEEE, 2018.

[7] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of stylegan," *CoRR*, vol. abs/1912.04958, 2019.

[8] M. C. Doukas, M. R. Koujan, V. Sharmanska, A. Roussos, and S. Zafeiriou, "Head2head++: Deep facial attributes re-targeting," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 3, no. 1, pp. 31–43, 2021.

[9] M. Cao, H. Huang, H. Wang, X. Wang, L. Shen, S. Wang, L. Bao, Z. Li, and J. Luo, "Task-agnostic temporally consistent facial video editing," *CoRR*, vol. abs/2007.01466, 2020.

[10] Y. Wei, Z. Gan, W. Li, S. Lyu, M.-C. Chang, L. Zhang, J. Gao, and P. Zhang, "Maggan: High-resolution face attribute editing with mask-guided generative adversarial network," in *Proceedings of the Asian Conference on Computer Vision*, 2020.

[11] Z. Xu, X. Yu, Z. Hong, Z. Zhu, J. Han, J. Liu, E. Ding, and X. Bai, "Facecontroller: Controllable attribute editing for face in the wild," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 3083–3091, 2021.

[12] A. H. Bermano, R. Gal, Y. Alaluf, R. Mokady, Y. Nitzan, O. Tov, O. Patashnik, and D. Cohen-Or, "State-of-the-art in the architecture, methods and applications of stylegan," in *Computer Graphics Forum*, vol. 41, pp. 591–611, Wiley Online Library, 2022.

[13] Y. Shi, X. Yang, Y. Wan, and X. Shen, "Semanticstylegan: Learning compositional generative priors for controllable image synthesis and editing," *CoRR*, vol. abs/2112.02236, 2021.

[14] Y.-J. Heo, Y.-J. Choi, Y.-W. Lee, and B.-G. Kim, "Deepfake detection scheme based on vision transformer and distillation," *arXiv preprint arXiv:2104.01353*, 2021.

[15] H. Zhao, W. Zhou, D. Chen, T. Wei, W. Zhang, and N. Yu, "Multi-attentional deepfake detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2185–2194, 2021.

[16] S. Li, V. Dutta, X. He, and T. Matsumaru, "Deep learning based one-class detection system for fake faces generated by gan network," *Sensors*, vol. 22, no. 20, 2022.

[17] S.-Y. Wang, O. Wang, R. Zhang, A. Owens, and A. A. Efros, "Cnn-generated images are surprisingly easy to spot... for now," 2020.

[18] S. Das, S. Seferbekov, A. Datta, M. S. Islam, and M. R. Amin, "Towards solving the deepfake problem: An analysis on improving deepfake detection using dynamic face augmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3776–3785, 2021.

[19] C. Wang and W. Deng, "Representative forgery mining for fake face detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14923–14932, 2021.

[20] S. Li, V. Dutta, X. He, and T. Matsumaru, "Deep learning based one-class detection system for fake faces generated by gan network," *Sensors*, vol. 22, no. 20, p. 7767, 2022.

[21] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401–4410, 2019.

[22] X. Zhang, S. Karaman, and S.-F. Chang, "Detecting and simulating artifacts in gan fake images," 2019.

[23] L. W. Sagers, J. A. Diao, M. Groh, P. Rajpurkar, A. S. Adamson, and A. K. Manrai, "Improving dermatology classifiers across populations using images generated by large diffusion models," 2022.

[24] L. C. Adams, F. Busch, D. Truhn, M. R. Makowski, H. J. W. L. Aerts, and K. K. Bressem, "What does dall-e 2 know about radiology?," *J Med Internet Res*, vol. 25, p. e43110, Mar 2023.

[25] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," 2021.

[26] C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman, *et al.*, "Laion-5b: An open large-scale dataset for training next generation image-text models," *arXiv preprint arXiv:2210.08402*, 2022.

[27] N. Pavlichenko and D. Ustalov, "Best prompts for text-to-image models and how to find them," *arXiv preprint arXiv:2209.11711*, 2022.

[28] Y. Hao, Z. Chi, L. Dong, and F. Wei, "Optimizing prompts for text-to-image generation," 2022.

[29] "Lit-LLaMA ." `https://github.com/Lightning-AI/lit-llama`, 2021.

[30] B. Dolhansky, J. Bitton, B. Pflaum, J. Lu, R. Howes, M. Wang, and C. C. Ferrer, "The deepfake detection challenge (dfdc) dataset," 2020.

[31] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "Faceforensics++: Learning to detect manipulated facial images," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1–11, 2019.

[32] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, "Celeb-df: A large-scale challenging dataset for deepfake forensics," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3207–3216, 2020.

[33] G. Bradski and A. Kaehler, "Opencv." `https://opencv.org`, 2021. Acesso em 29 de maio de 2023.

[34] Google, "Mediapipe." `https://mediapipe.dev`, 2021. Acesso em 29 de maio de 2023.

[35] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," *CoRR*, vol. abs/1708.04896, 2017.

[36] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[37] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*. O'Reilly Media Inc., 2009.

[38] Y. Shi, X. Yang, Y. Wan, and X. Shen, "Semanticstylegan: Learning compositional generative priors for controllable image synthesis and editing," *CoRR*, vol. abs/2112.02236, 2021.

[39] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in vision: A survey," *ACM Comput. Surv.*, vol. 54, sep 2022.

[40] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," *CoRR*, vol. abs/2012.12877, 2020.

[41] R. Wightman, "Pytorch image models." `https://github.com/rwightman/pytorch-image-models`, 2019.