O Uso de IA Para Detecção de *Deepfakes* em Mídias Sociais

Pedro Henrique Novicov de Andrade[1], **André Rodrigues Oliveira**[1]

[1]Faculdade de Computação e Informática (FCI)
Universidade Presbiteriana Mackenzie, São Paulo-SP, Brasil.

Resumo. *Este projeto visita a história de detecção de vídeos e imagens criadas por algoritmos de Inteligencia Artificial, conhecidas comumente como "deepfakes". Dentre as contribuições, foram realizadas estimativas de quando um indivíduo comum poderá criar um modelo capaz de gerar deepfakes no mesmo nível das ferramentas feitas pelas grandes companhias, baseados em quantidade de armazenamento e poder de processamento necessários; também testamos alguns modelos propostos para detecção de deepfakes com diferentes datasets, mostrando quais modelos possuem mais chances de serem generalizados. Com base na pesquisa realizada, fornecemos recomendações de hardware e para criação de modelos para futuros trabalhos na área detecção.*
Palavra-Chave*: inteligência artificial; detecção de deep fakes; Redes generativas adversariais; modelos de difusão.*

Abstract. *This project explores the history of video and image detection created by Artificial Intelligence algorithms, commonly known as "deepfakes."Among the contributions, we made estimates of when an ordinary individual will be able to create a model capable of generating deepfakes at the same level as the tools made by large companies based on the amount of storage and processing power required; we also tested some proposed models for deepfake detection with different datasets, showing which models have the best chance of being generalized. Based on the research carried out, we provide hardware recommendations for creating models for future work in the detection area.*
Keyword*: artificial intelligence; deep fake detection; generative adversarial networks; diffusion models.*

1. Introdução

Recentemente, há um aumento no número de vídeos, imagens e áudios modificados ou completamente criados por Inteligências Artificiais (IA), sendo que o Brasil teve um aumento de 830% no numero de *deepfakes* de 2022 a 2023 (Sacramento 2024). Estes vídeos variam de presidentes americanos rivais jogando *vídeogames* a criadores de conteúdo tendo seus semblantes colocados em pornografia. Mesmo que no estágio atual consigamos perceber *deepfakes* com um certo grau de facilidade, com o avanço da tecnologia esta facilidade tenderá a diminuir. Uma ferramenta capaz de verificar se uma foto ou vídeo é real pode salvar a reputação de pessoas inocentes e impedir a propagação de notícias falsas, um fato enfatizado pelo estado reacionário das mídias sociais (UFJF 2023). Adicionalmente, mesmo que as ferramentas mais famosas proíbam a criação de conteúdo capaz de causar prejuízo a terceiros isto não impede pessoas má-intencionadas de gerar e criar fotos ou vídeos utilizando seus próprios modelos.

Este problema não está sendo ignorado pelos fabricantes das ferramentas de geração de imagem como o Google, responsável pela ferramenta *Gemini* (Team et al. 2024),

tendo delineado 6 elementos de segurança no uso de inteligência artificial (Google 2023). Porém, mesmo que companhias criem métodos de segurança para suas ferramentas, isto não impede que pessoas comuns criem seus próprios modelos localmente, ainda que seja difícil estabelecer um comparativo entre a capacidade de um indivíduo e de uma corporação como o Google em relação a *deepfakes*, visto que na documentação pública destas ferramentas não há número de parâmetros nem muitos detalhes da implementação (Imagen-Team-Google et al. 2024). Porém, ainda é possível realizar uma estimativa para averiguar a capacidade de um indivíduo criar um modelo com o mesmo poder que aqueles feitos por grandes organizações. Além disso, é importante ressaltar que já existem *deepfakes* de boa qualidade feitas por indivíduos e pequenas organizações. Por exemplo, existe um site que vende *deepfakes* pornográficas baseadas em *influencers* que fazem lives no site Twitch; e pelo menos no Estados Unidos isto não é ilegal o suficiente para derrubar o site em questão (Britton 2023), um fato que certamente é um incômodo para as pessoas que tiveram sua imagem colocada em situações constrangedoras sem sua permissão. Um conceito importante quando falamos em detecção de imagens fakes é o *Uncanny Valley*, mostrado na Figura 1.

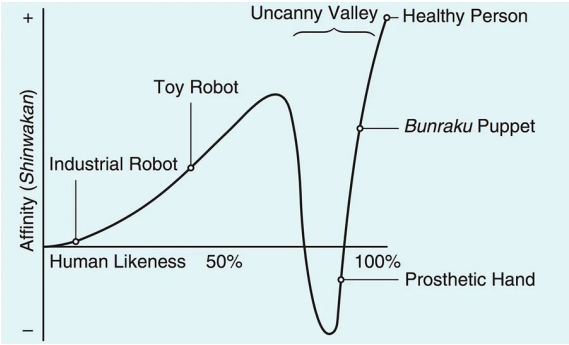

Figura 1. O *Uncanny Valley* como definido no artigo original

O *Uncanny Valley* (Mori et al. 2012), como definido por Masahiro Mori em 1970, é a súbita queda da afinidade de um indivíduo com um item quando ele chega perto o suficiente da forma humana, conforme o gráfico da Figura 1. Alguns exemplos conhecidos seriam: o filme "O Expresso Polar"de 2004, manequins e Bonecas de Porcelana. Uma pesquisa recente de 2023, chamada de "*Have We Crossed the Uncanny Valley? Understanding Affinity, Trustworthiness, and Preference for Realistic Digital Humans in Immersive Environments*" (Seymour et al. 2021), foi feita para determinar se já passamos deste ponto. Ela detalhou que ao entrevistar múltiplas pessoas com 3 avatares de diferentes qualidades em tanto um ambiente de realidade virtual quanto por um monitor normal, foi encontrado que, para algumas pessoas, não houve estranheza. Portanto, se já passamos do ponto onde algumas pessoas não se sentem desconfortável com o artificial, quando chegaremos ao ponto onde a distinção entre real e o falso será imperceptível? Isto é exacerbado pelas consequências mórbidas de algoritmos criados por Inteligências Artificiais falando como seres humanos, como no caso onde um jovem 14 anos cometeu suicídio devido a suas conversas com um *chatbot* de acordo com sua mãe (Duffy 2024).

1.1. Objetivos

Listamos os objetivos gerais deste trabalho a seguir:

- Entender o estado atual da criação de *deepfakes*;
- Averiguar os requerimentos computacionais para criar modelos de ponta atuais;
- Comparar o hardware comercial voltado ao público geral com o hardware feito para pesquisa, para então quantificar quando um cidadão comum terá hardware capaz de criar uma *IA* no mesmo nivél de grandes companhias sem assistência;
- Entender as peculiaridades da detecção de *deepfakes*, observando os resultados atuais e como foram encontrados;
- Clarificar pontos deixados em aberto por outros trabalhos;
- Averiguar se há diferença na detecção de *deepfakes* de seres humanos com o resto do mundo natural;
- Delinear pontos para se manter em mente e direcionar futuras tentativas e implementações de detecção de *deepfakes*.

1.2. Termos e Conceitos

Inteligências artificiais (IA), ou mais especificamente modelos de aprendizagem de máquina, são algoritimos que se modificam ou aprendem com base em dados fornecidos (Trindade and Oliveira 2024). Um conjunto de dados é chamado de *dataset*. Modelos de aprendizado profundo, aqui abreviados para o termo mais geral *IA*, são compostos de diversas camadas, sendo que cada camada representa uma transformação sobre os dados recebidos. Após a construção da estrutura do modelo, há o período de aprendizado da *IA*, que fará ajustes em seu algorítimo de acordo com algumas funções pré-estabelecidas, sendo elas: função de perda, que é baseada na diferença da predição com a verdade; e a função otimizadora (otimizador), que utiliza variáveis primariamente estáticas. Uma *IA* normalmente percorre os dados definidos para o treinamento várias vezes. Cada percurso pelos dados pode ser chamado de eras, épocas ou iterações. Não há um número concreto de eras necessários para que um modelo alcance uma boa precisão, mas, a partir de certo ponto, a universalidade do modelo começa a diminuir, com o algoritmo se especializando nos dados de treinamento. Esse fenômeno é chamado de *overfit*. Neste trabalho diversos tipos de modelos serão abordados. Para fazer-se uma desambiguação, eles serão definidos como:

- *Foundation Model*, ou um modelo de fundação, são modelos treinados com dados gerais e capazes de fazer múltiplos tipos de atividades (Bommasani et al. 2022). Modelos como chatGPT e CLIP caem neste campo. Os modelos mencionados futuramente poderão estar nesta categoria; Porém, o foco deste trabalho está na sua capacidade de criação de imagem, então, não será mencionado suas outras capacidades a não ser que elas sejam relevantes.
- *Large Language Model* (*LLM*), cuja tradução é modelo grande de linguagem, é um tipo de aprendizado de máquina focado no processamento de dados envolvendo a linguagem humana (Naveed et al. 2024). Este tipo de modelo é usado com modelos geradores para interpretar o comando do usuário na criação de imagens (Team et al. 2024).
- *Generative Adversarial Network* (*GAN*), cuja tradução seria Rede Generativa Adversarial, é um modelo de aprendizagem composto por duas *IA*s, uma para gerar o conteúdo e outra que discrimina o conteúdo criado (Goodfellow et al. 2014). Os dois algoritmos vão se aprimorando pelas eras de treinamento. Este foi o modelo mais dominante para a geração de imagens por volta de 2020 (Kong et al. 2020).

- *Diffusion Model*, ou modelos de difusão, são modelos de aprendizado de máquina em que o ruído é introduzido e depois retirado da imagem/vídeo original. A introdução do ruído é feita por um Kernel de Transição Gaussiano e a remoção por uma função variável (Chen et al. 2024). Este é o modelo usado por ferramentas de geração como o *Imagen* da Google (Google 2024) e *DALL-E 3* (Betker et al. 2024).
- *Text-To-Image Model* (Face 2024), ou modelo de texto para imagem, é um nome usado para modelos de criação de imagem que utilizam o input de um usuário para criar uma imagem.
- Modelos de classificação de imagem (Wang and Su 2019) também serão chamados de detectores neste trabalho; eles utilizam os dados analisados para categorizar dados gerais. Na classificação de vídeos e imagens é possível utilizar *labels*, também chamados de etiquetas. Etiquetas contém dados gerais sobre o *dataset*, que podem ser: caminho interno dos arquivos pelo armazenamento, a classe do arquivo e qualquer informação que possa ser útil, como por exemplo a localização de boca e orelhas para um modelo que seja centrado em faces humanas.

1.3. Tentativas Prévias de Detecção de *Deepfakes*

O potencial perigoso das *deepfakes* já foi notado, porém muitos destes exemplos estão desatualizados dados os avanços no campo de criação de *deepfakes*. Ainda assim, eles ainda são úteis em termos de direção e fornecimento de dados. Obter os dados necessários para treinar detectores é uma tarefa desafiadora, exigindo esforços significativos tanto humanos quanto computacionais. Além disso, há diversas restrições para a aquisição de imagens geradas, incluindo barreiras monetárias, geográficas e limitações impostas pelas ferramentas. Outra dificuldade, é a falta de centralização de informações sobre a detecção de *deepfakes*, sendo que o melhor lugar para se entender o estado atual é um repositório no *GitHub*, mantido por um grupo de três pessoas (Zhang et al. 2024).

Há várias maneiras de se detectar *deepfakes*. Este trabalho as separa em 3 categorias com base em seu escopo:

- Generalizadores, onde diferentes técnicas são utilizadas, para que o modelo não faça *overfit* em um único modelo e possa ser utilizado por dados de múltiplos modelos. Como exemplo temos técnicas como integrar a profundidade de RGB (Leporoni et al. 2024) e a criação de um gerador de imagens para servir como fonte secundária de dados;
- Meta, onde os detectores procuram detalhes mais específicos como a homogenia entre voz e face (Cheng et al. 2022), a utilização de um transformador espaço-temporal (Zhang et al. 2022) e atribuição ao modelo gerador (Jia et al. 2022)
- Preventivos, que procuram facilitar a detecção por métodos externos, como por exemplo a atribuição de uma "marca-d'água"a foto original (Yang et al. 2021) e ataques adversários que atrapalhariam a criação da *deepfake* (Dong et al. 2022).

Dois trabalhos em especifico disponibilizaram seus dados para uso público e terem inspirado em parte a metodologia deste trabalho: o *CCNSpot* (Wang et al. 2020) e o *GenDet* (Zhu et al. 2023). O *CCNSpot* (Wang et al. 2020) publicado em 2019, explora a detecção de *deepfakes* e obteve bons resultados com 90% de precisão dentro de seu *dataset*. Porém, quando testado pelo *GenDet* (Zhu et al. 2023) ele apresentou os piores resultados quando comparado aos modelos genéricos utilizando modelos de classificação

de imagem existentes. Isto não significa que seus resultados foram completamente invalidados, ainda mais levando em consideração que este trabalho foi feito antes da popularização de modelos de difusão para a criação de imagens e, portanto, utiliza primariamente *GANs*. Seus dados de treinamento e validação foram incorporados em um trabalho subsequente chamado *Universal Fake Dataset* (Ojha et al. 2024) que será usado parcialmente na metodologia deste trabalho.

O *GenDet* (Zhu et al. 2023), foi uma ótima fonte de dados pois, enquanto a proposta dele centra em seu próprio detector, seus autores detalham testes envolvendo a capacidade de modelos pré-peparados como o *resnet* (He et al. 2015) em detectar *deepfakes*. Para referência, o detector proposto utilizou um gerador de imagens interno para generalizar os dados recebidos, como uma *GAN* invertida. Ele é um trabalho recente, publicado no final de 2023, e já é bem popular em termos de citações. Ele disponibiliza um volume alto de dados, com o *dataset* sendo separado por gerador e depois subdividido em uma parte de treinamento, que consiste de 320 mil imagens, e uma parte de validação, que consiste de 12 mil imagens. O número de imagens reais e falsas dentro de cada parte são similares. Os testes disponibilizados foram feitos com detectores treinados que utilizaram o *Stable Diffusion* versão 1.4 (*SDV1.4* (Rombach et al. 2021)) como fonte e detectores que usaram uma combinação de todos os dados como fonte. Algumas observações importantes para se fazer sobre os dados disponibilizados são: há dois lugares que os criadores deixam disponível o *dataset*, mas só um deles é acessível devido ao segundo ser acessível apenas dentro do território Chinês; não foi disponibilizado a implementação do detector mencionado pelo trabalho, portanto não foi possível verificar a efetividade real de seu gerador interno como fonte de dados secundária para o detector; o trabalho disponibiliza dados de um gerador que eles chamam de *Wukong*, mas não foi achado nenhum detalhe deste gerador; como o link do gerador leva ao mesmo site com restrições para fora da China não foi possivel verificar se isto é uma restrição ou se o link original está morto ou incorreto.

2. Estimativas Sobre a Criação de Modelos Generativos

O objetivo desta seção é averiguar o quanto de memória (armazenamento) e processamento é necessário para construir um modelo gerador parecido como as de grandes empresas. Procuramos as especificações técnicas dos modelos generativos disponíveis, porém, há uma falta de informações técnicas disponíveis. Portanto, as estimativas são substanciadas, mas não concretas.

2.1. Armazenamento

Começando pela questão do armazenamento e disponibilidade dos dados necessários para treinar um modelo capaz de *deepfakes*. Placas Mãe modernas possuem entre 4 a 10 entradas do tipo SATA e 1 a 2 conectores do tipo NVMe, então, com discos rígidos comercialmente disponíveis tendo uma capacidade de 0.5 TB até 20 TB, e SSDs do tipo NVMe tendo capacidade de 0.256 TB até 4 TB, um indivíduo com apenas uma máquina usando todos os conectores pode ter entre 2.256 TB a 208 TB. Para adquirir os dados necessários não haveria dificuldades em termos de disponibilidade, já que, durante está pesquisa, foi encontrado uma grande quantidade material pré-preparados para treinamento de modelos na forma de mais de uma centena de *datasets* focados em seres humanos. Para dados frescos e não pré-preparados, a rede social Meta (Facebook), de acordo com si mesma, recebe em torno de 4 PB(petabytes) por dia e tem 300 PB salvos

em sua base de dados (Wiener 2024). Então, mesmo que em torno de 1% deste material diário seja utilizável para treinamento de modelos geradores, um indivíduo ainda teria acesso a 40 Terabytes de material por dia. Com uma ideia da capacidade de armazenamento e do volume de material disponível *online*. Foi feita a estimação do espaço requerido para o material de treinamento. Utilizando, um número arbitrário, 80 para o número de classes, que é mesmo número no *dataset* COCO (Lin et al. 2015), que é um *dataset* de segmentação de objetos patrocinado por múltiplas companhias, incluindo Microsoft e Facebook. Com, a medida de 5 mil itens por classe de treinamento como o mínimo necessário para criar um modelo. Então seria necessário por volta de 400 mil imagens e com cada imagem tendo em média 1.2 MB (1.2 MB é a média do tamanho de uma foto de instagram (Lindner 2024)) resultaria em um espaço de 0.48 TB de material para criar um modelo simples. No caso onde 1 milhão de imagens são utilizadas por categoria, o armazenamento final seria de 96 TB. Também é bom ressaltar que, os modelos gerativos que aceitam o uso da linguagem humana como *input* do usuário necessitam que os dados de treinamento venham com uma frase de referência. Para criar a estimativa do espaço tomado por este tipo de dado. Foi usado como referência os dados do *dataset Flickr Diverse Faces* (fdf) (Hukkelås et al. 2019), que contém em torno de 7.35 milhões de imagens e toma 64 GB de espaço no disco. Ele tem um arquivo em formato JSON contendo sua *metadata* que toma entorno de 1.7 GB de espaço, e este número vai para 2.65 GB quando é incluído as informações extras sobre as imagens. Isto resulta em uma média de 1 GB de informação extra para cada 2.77 milhões de imagens ou 360,24 bytes para cada imagem.

2.1.1. Processamento

Agora, para estimar a capacidade de processamento e o tempo necessário para criar um modelo de ponta. Dentre as peças de um computador, as mais significativas são: o processador e a placa de vídeo (GPU)(Systems 2024), com a placa de vídeo sendo a peça mais limitante e o maior foco desta subseção. Um ponto a se fazer é que, no momento, a Nvidia, uma das três grandes companhias que fazem placas de vídeo, é a mais focada em Inteligência Artificial (Intelligence 2024); tanto que a biblioteca pytorch, que constitui a *backend* de muitos modelos requer placas da Nvidia para suas versões principais (Pytorch 2024), portanto este trabalho foca somente de placas de vídeo da Nvidia para estas estimativas e comparações. As métricas mais relevantes são: a memória de acesso aleatório visual (*VRAM*) e a sua taxa de transferência, que é a métrica de quantos dados são transferidos por segundo. Há também uma métrica chamada *memory bus* que indica a quantidade de caminhos utilizáveis para transferência de dados simultânea; não é claro o efeito desta métrica em especifico dado que, durante a comparação das placas de vídeo, A100 e *GeForce 4090* (4090), foi encontrado uma diferença de 36% na taxa de transferência para uma disparidade de 1333% no *memory bus*. Para servir de ponto inicial, foi utilizado o artigo "How We Trained Stable Diffusion for Less than $50k (Part 3)" (Patel et al. 2023), onde é relatado como foi recriado o modelo de difusão chamado *stable diffusion* 2.0 (Rombach et al. 2021) por menos de 50 mil dólares. Os autores utilizaram placas de vídeo A100 e fizeram optimizações no processo de treinamento resultando em uma melhoria de 2.71 vezes e notaram uma relação linear entre o número de placa de vídeos e o tempo necessário para fazer o treinamento. A placa A100 da Nvidia

utilizada pelo artigo tem 40 GB de *VRAM*, uma *memory bus* de 5210 bit e uma taxa de transferência de 1.56 TB/s. Em testes com 8 dessas placas e com imagens de 256x256 foi alcançado uma velocidade de 1100 imagens por segundo e levaria 101.04 dias, enquanto com 128 placas tendo uma velocidade de 11600 imagens por segundo e durou 6.79 dias. Uma placa da Nvidia 4090 possui um VRAM de 24 GB e uma taxa de transferência de 1.01 TB/s, então entre a A100 e a 4090 há uma diminuição de 40% para a VRAM e 36% na taxa de transferência. Uma disparidade grande, mas, quando olhamos as estimativas para a próxima geração de placas de vídeo comercial da Nvidia, que serão lançadas no começo do ano de 2025, vemos VRAM de 32 GB e taxa de transferência de 1.42 TB/s; o que é uma disparidade de 20% e de 9% respectivamente, quando comparada a A100. Mesmo que estes dados não estejam precisos, um aumento nessas duas métricas é certa.

Portanto, dado a capacidade das peças de hardware relevante, um indivíduo comum não conseguiria criar um modelo gerativo com *hardware* local. Mesmo com o armazenamento não sendo problemático, dado que as diversas formas de adquirir e guardar dados e maneiras de transferir o aprendizado feito com estes dados para um outro modelo. A questão de armazenamento é mais sobre conveniência do que habilidade, enquanto que, na área de processamento, o problema não é facilmente resolvido. Já que, mesmo com 128 placas de vídeo voltadas para este tipo de processamento demoram quase uma semana para processar todos os dados necessários, um indivíduo não conseguiria treinar um modelo em tempo razoável com placas com menos placas com piores especificações relevantes, ou seja, mesmo que as placas de vídeo comerciais cheguem nas mesmas especificações de suas placas irmãs, o custo para conseguir mais 127 placas e integrá-las em um sistema local seria inviável para uma pessoa comum. Então, para criar um modelo com *hardware* comercial, precisaria de uma rede de computadores robusta com mais de uma centena de máquinas trabalhando em conjunto, com peças de ponta. Há a possibilidade de que um indivíduo utilize métodos dúbios como utilizar a placa de vídeo de outra pessoa em segredo, algo que já foi feito para a mineração de crypto-moedas (Radulovic 2018), o uso excessivo e desconhecido da placa seria óbvio e facilmente descoberto. Não é impossível que uma nova companhia entre no mercado disponibilizando placas de vídeo focadas nesses problemas, mas, no momento, não foi identificado nenhum sinal de interesse público para isso ser viável, especialmente dado que placas comerciais já são capazes de gerar imagens rapidamente, vide figura 2, onde até mesmo a pior placa tem uma performance de 12.64 imagens por minuto. Portanto, salvo uma grande descoberta que, facilite o processamento de dados, ou melhore drasticamente as habilidades de *GPUs*, não há como um indivíduo criar um modelo gerador de alto calibre sem ajuda financial. Para referencia, a placa A100 foi lançada com um preço de 8 mil dólares e não teve reduções em seu preço desde seu lançamento em 2020. Isto é comum dado que placas de vídeos tendem a receber reduções em seu preço após o lançamento de uma nova geração na mesma linha, com a placa de vídeo prévia, a V100, tendo seu preço de lançamento de 10 mil dólares para uma média de mil dólares. Então com os algorítimos utilizados hoje, e dado que não tenha uma grande mudança no treinamento de modelos geradores, uma pessoa não conseguiria criar um modelo gerador que nem as feitas por grandes empresas enquanto não houver uma placa de vídeo com o quadruplo da capacidade da placa A100 com um preço acessível. Pois, com está placa teorética seria necessário apenas 16 placas de vídeo para realizar o treinamento em torno de 12 dias. É relevante lembrar que modelos de qualquer tipo não precisam ser treinados completamente de uma vez, isso quer dizer

que é possível treinar por um certo tempo e fazer uma pausa para manutenção e evitar desgaste de hardware. Antes de seguir para a parte de explicação de detectores, gostaria de ressaltar que, já são criadas *deepfakes* caseiras, e, que o gargalo na criação de vídeos por modelo geradores(Li et al. 2024) não durarão para sempre. Principalmente, quando há novas ferramentas sendo criadas para facilitar a utilização da Nuvem no treinamento de modelos, para referência, o artigo que reporta o treinamento do *Stable Diffusion 2* por menos de 50 mil dólares utilizou a nuvem.

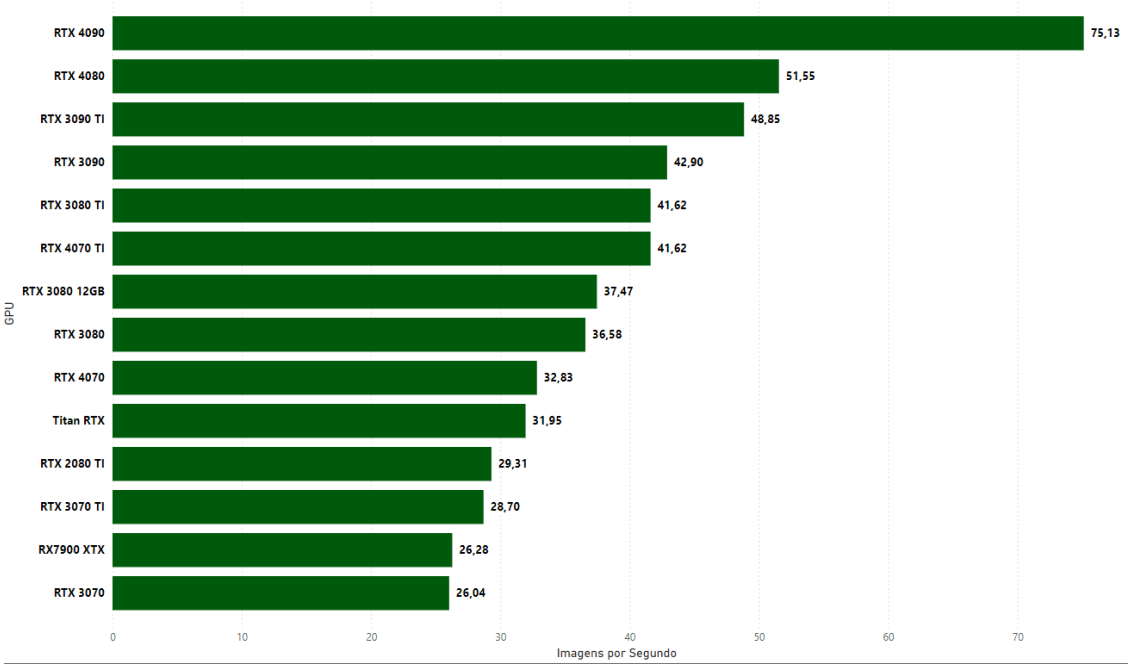

Figura 2. Versão simplificada dos dados apresentados pela publicação *Tom's Hardware* (Walton 2023)

3. Metodologia

3.1. Ferramentas e Dados

Para o processamento e análise de dados foi se utilizado a linguagem de programação Python, por ser a linguagem mais popular para análise de dados (of Dublin 2024). Python também possui acesso as duas bibliotecas de aprendizado de máquina, sendo elas TensorFlow (Abadi et al. 2015) e Pytorch (Paszke et al. 2019). Em termos de hardware foi utilizada uma placa de vídeo da Nvidia modelo 3050 com 8 GB de VRAM junto com um processador da 11ª geração da Intel, modelo i7-11700F com *overclock*. Na questão de dados utilizados para o treinamento dos detectores e seus subsequentes testes de performance, foi utilizado uma variedade de *datasets* neste trabalho, aqui estão o seus nomes com uma breve descrição deles:

- Flickr-Faces-HQ Dataset (FFHQ): um *dataset* contendo 70 mil imagens de faces reais recolhidos da plataforma Flickr.
- *GenImage* (Zhu et al. 2023): Dataset, mencionado anteriormente, que contém mais de 1 milhão de imagens, separadas por modelo e veracidade. Ele consiste dos modelos *ADM* (Dhariwal and Nichol 2021), *BigGAN* (Brock et al. 2019),

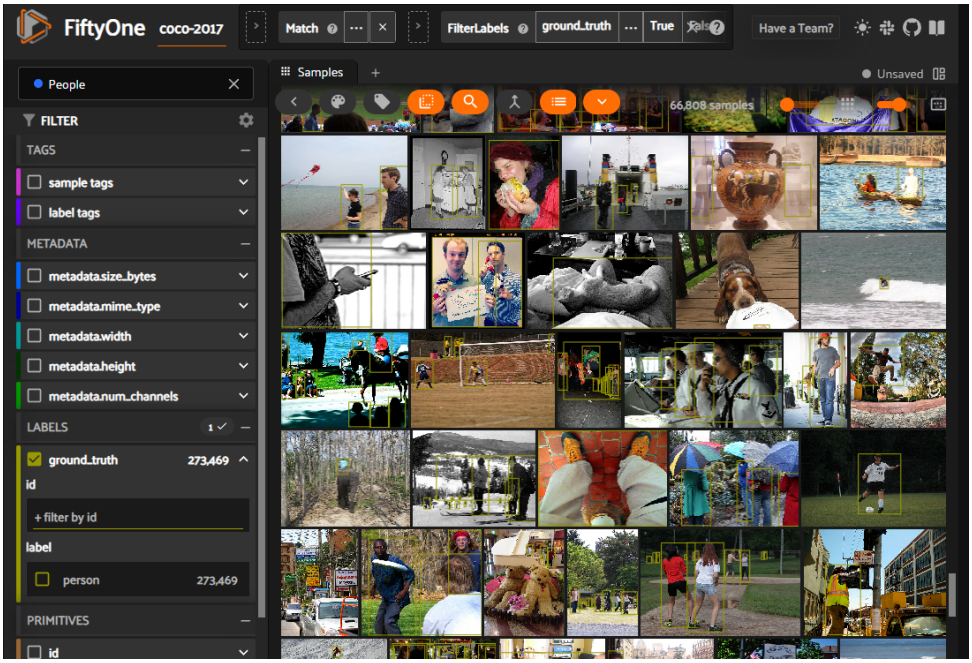

Figura 3. *Voxel 51* com as imagens do dataset COCO

VDQM (Gu et al. 2022), *Stable Diffusion V1.4(SDV1.4)* (Rombach et al. 2021), *Glide* (Nichol et al. 2022), *Midjourney* (Midjourney 2024) e *Wukong*

- *Generated Photos* (Photos 2024): *dataset* de 10 mil faces artificiais sem plano de fundo. Em seu site não é especificado nada além de que o modelo utilizado ser uma *GAN*
- *Universal Fake Dataset* (Ojha et al. 2024): *dataset* que herdou os dados do*CCNSpot* (Wang et al. 2020). Deste *dataset* utilizamos as imagens dos modelos *StarGan*(Choi et al. 2018), *StyleGan* (Karras et al. 2019) e *CycleGan* (Zhu et al. 2020)
- *DFFD: Diverse Fake Face Dataset* (Dang et al. 2020): *dataset* focado em faces humanas. Dos dados provenientes dele foram utilizados *PgGan_v2* (Karras et al. 2018) e *StyleGan* (Karras et al. 2019)

Observação, para diferenciar os dados de *StyleGan* (Karras et al. 2019) vindo de dois *datasets* diferentes, o *StyleGan* (Karras et al. 2019) proveniente do *DFFD* será referido como *StyleGan_ffhq*.

3.2. Estrutura

Os dados foram explorados e manipulados utilizando o programa *Voxel 51*, por ser um método fácil de se implementar com métodos de integração com as bibliotecas de criação de inteligência artificial. Também, após os dados serem limpos e compreendidos foi possível retira-lo, facilmente devido a maneira em que os dados foram utilizados. Após a exploração de dados, esses mesmos dados foram transformados para uma forma apropriada ao modelo. O modelo foi treinado com a utilização das imagens filtradas pela ferramenta, e seria subsequentemente avaliada, com os resultados sendo colocados em um arquivo tabulado por meio de código.

3.3. Testes Preliminares

Os primeiros teste feito para este trabalho foram feitos para se familiarizar com as bibliotecas usadas para o aprendizado de máquina,, e a importação dos dados. Com estes testes ficou evidente rapidamente a tendência de detectores se especializarem no modelo em que são treinado com as primeiras iterações dos algorítimos possuindo precisões de no mínimo 82% no teste de validação. A única exceção foi quando treinando com um modelo simplístico com os dados do *midjourney*, o que é o maior dos conjuntos, em termos de armazenamento(212 GB) - logo após essa primeira iteração os resultados foram semelhantes aos outros modelos em relação aos seus próprios dados de validação.

Foram treinados 2 tipos de modelos, utilizando imagens de diferentes modelos fontes e testadas em imagens de outros modelo, e com os seus resultados sendo colocados em uma tabela. Os 2 modelos do teste preliminar são: Simples, nomeado assim devido a utilizar o exemplo da biblioteca *TensorFlow* (Abadi et al. 2015) com algumas técnicas de melhoria de dados delineadas na mesma página, seus resultados estão listados na tabela 1. O Simples utilizou o otimizador *AdamW* com a perda sendo definida pela função *SparseCategoricalCrossentropy* do *TensorFlow* (Abadi et al. 2015) que funciona de acordo com esta equação (Arat 2018):

$$L(y_t, y_p) = -\frac{1}{B}\sum_{b=1}^{B}[y_b^t log(y_b^p) + (1 - y_b^p) log(1 - y_b^p)]$$

Na fórmula acima, L denota a perda resultante, y_t e y_p significa a classificação real e a classificação resultante da máquina respectivamente, B denota o total de amostras (imagens) sendo analisadas de uma vez, b significa a amostra atual. Para referência este modelo tem uma precisão média de 60% no exemplo original para um dataset de identificação de flores. Suas camadas são: camada de melhoramento de dados, onde dados de treinamento são levemente mudados para evitar *overfit*. Depois, vem três pares de camadas de convolução 2D e *maxpooling2D*, a camada de convolução extrai a informação por meio de filtros (kernels), com cada filtro pegando uma informação diferente; a camada *maxpooling2D* é responsável por manter somente as informações relevantes. A camada seguinte é chamada de *droppout*, ela desliga uma porcentagem do neurônios do modelo, que neste caso seria 0.2, ou 20%. Agora, tudo passa para um processo de "achatamento", onde os dados são preparados para serem processados pela camada densa. A camada densa, composta por 128 neurônios, combina as informações recebidas para inferir padrões complexos e as enviar para a camada classificadora. Por último, a camada classificadora gera a previsão final baseada no que recebeu. Um diagrama visual deste modelo se encontra na figura 4.

Podemos ver que houve uma anomalia quando o Simples foi treinado com os dados de *VDQM*(Gu et al. 2022). Dado que há uma variância nos testes de validação pós-treinamento, é possível que os dados de treinamento do *VDQM* tenham sido misturados com os dados de outros modelos. Para confirmar está suspeita, foi feita uma comparação entre os resultados do Simples com os resultados de um modelo de detecção mais complexo. Este modelo foi o *resnet50*(He et al. 2015). Este modelo foi utilizado por dois motivos: ele foi um dos modelos utilizados pelo *GenDet*(Zhu et al. 2023), então foi possível ter uma base para performance do modelo; o outro motivo sendo que há implementações oficiais desta estrutura no TensorFlow. Foi usado os mesmos algorítimos de perda e otimização do Simples. Seus resultados estão listados na tabela 2. Também foi feito um diagrama visual que está na figura 5.

Simples	*adm*	*Wukong*	*SDV1.4*	*SDV1.5*	*BigGan*	*Midjourney*	*Glide*	*VDQM*
adm	99%	50%	50%	50%	51%	50%	75%	57%
Wukong	46%	93%	91%	91%	46%	56%	48%	51%
SDV1.4	51%	95%	96%	95%	51%	59%	52%	54%
BigGan	50%	50%	50%	50%	88%	50%	64%	52%
Midjourney	48%	70%	74%	74%	40%	82%	50%	43%
Glide	75%	50%	50%	50%	52%	50%	93%	51%
VDQM	65%	49%	49%	49%	83%	51%	81%	79%

Tabela 1. Tabela de performance do modelo simples. As linhas são os dados de treinamento e colunas são dados de de validação. Números foram arredondados

Resnet50	*ADM*	*Wukong*	*SDV1.4*	*SDV1.5*	*BigGan*	*Midjourney*	*Glide*	*VDQM*	Avg
ADM	99%	50%	50%	50%	50%	50%	50%	50%	56%
Wukong	49%	98%	96%	96%	49%	52%	50%	50%	68%
SDV1.4	49%	96%	96%	96%	48%	54%	51%	49%	67%
BigGan	52%	50 %	50%	50%	99%	50%	75%	56%	60%
Midjourney	48%	63%	63%	63%	49%	82%	51%	50%	59%
Glide	59%	50%	50%	49%	95%	50%	99%	66%	65%
VDQM	64%	50%	50%	50%	58%	51%	66%	99%	61%

Tabela 2. Resultados do modelo de detecção feito com o *resnet50* (He et al. 2015). As linhas são os dados de treinamento e colunas são dados de de validação.

Agora é possivel ver que os resultados do treinamento com dados vindo do *VDQM* (Gu et al. 2022) estão mais em linha com os resultados anteriores, enquanto ainda mantendo um número mais alto de acertos para os modelos *ADM* (Dhariwal and Nichol 2021) e *Glide* (Nichol et al. 2022) quando comparados ao resto dos modelos. A tendência da precisão ficar em 50% quando os dados foram compostos por imagens provenientes de modelos desconhecidos, relativo as imagens de treinamento, continua. Para entender melhor este comportamento dos detectores, foi separado um conjunto de dados composto de imagens propositalmente desbalanceadas com 351 fotos reais e 39 fotos falsas para ver como o modelo está lidando com modelos desconhecidos, ou seja 90% de imagens verdadeiras e 10% de imagens falsas. Em sequencia foram acrescentadas 146 imagens falsas para o dataset, mudando proporção de 65% real para 35% falsas. Isto resultou em uma precisão média de 80% e 60% em ordem. Este comportamento se manteve quando mudamos os modelos. Com base nos testes, podemos afirmar que o detector tende a errar para falso negativos.

3.4. Inter-Compatibilidade e as Possíveis fraquezas de Modelos complexos

Pelos dados expostos, nas tabelas 1 e 2, um detector tem uma facilidade em detectar certos modelos gerativos dependendo do modelo gerativo que providenciou as imagens de referência. Essa facilidade depende das estruturas originais como *ADM* (Dhariwal and Nichol 2021) e *Glide* (Nichol et al. 2022) ambos se declaram *Guided Diffusion Models* (modelos de difusão guiados). Com este conceito em mente olha-se brevemente para o *Wukong*. Como mencionado anteriormente, não há detalhes acessíveis sobre ele, mas considerando os resultados exibidos na tabela, pode-se ver que ele é similar aos modelos *SDV1.4* e *SDV1.5* (Rombach et al. 2021). Este grau de semelhança entre

diferentes modelos gerativos será referida como inter-compatibilidade, IC e para dar uma forma concreta a esta métrica usara-se a diferença dobrada da precisão média do detector com as imagens de um gerador *X* utilizado no treinamento (p_X), com a precisão média com imagens vindo de um gerador desconhecido *Y* p_Y. Ou seja:

$$IC(p_X, p_Y) = (p_X - p_Y) * 2$$

Essa equação é proposta para servir como referência, permitindo avaliar a semelhança entre os dados gerados por diferentes modelos e identificar eventuais falhas no detector. Por exemplo, se o valor de IC exceder 100, isso pode indicar erros estruturais no detector que comprometem a sua universalidade.

Como exemplo, observando a tabela 3, com os resultados do modelo *ResNet50* (He et al. 2015) treinado com os dados da *BigGan* (Brock et al. 2019) e *ADM* (Dhariwal and Nichol 2021), vê-se que o detector treinado com dados do *BigGan* (Brock et al. 2019) tem uma vantagem tremenda sobre o detector treinado com o *ADM* (Dhariwal and Nichol 2021) nos testes com imagens do *StarGan*(Choi et al. 2018).

	StarGan	*StyleGan*	*StyleGan_ffhq*	*CycleGan*	*PgGan_v2*
BigGan	97%	51%	52%	50%	50%
ADM	37%	50%	50%	50%	50%

Tabela 3. Dados de validação, a proporção dos dados vindo da*StarGan* não são proporcionais, com, aproximadamente, 37% sendo verdadeiras e 63% falsas.

3.5. Testes Intermediários

Com uma ideia sólida da capacidade de detecção de *deepfakes* de um modelo treinado com imagens feitas por um modelo gerador, foi testada a performance do modelo com uma união dos dados vindos dos modelos: *Midjourney* (Midjourney 2024), *BigGan* (Brock et al. 2019) e *Glide* (Nichol et al. 2022). Não há detalhes sobre a estrutura do *Midjourney* (Midjourney 2024) mas se olhando o *IC* ele é similar ao *SDV1.4* (Rombach et al. 2021), Wukong. O *VDQM* (Gu et al. 2022) também é elegível a ser colocado no conjunto de treinamento. Ele vai ser excluído para ser usado como teste da universalidade do detector. Também foi observado oque acontece com o *IC* quando foi feito o treinamento utilizando o conjunto.

A primeira variante usa a arquitetura **Professor-Aluno(P&E)**. Também conhecida como *Distilation Learning*, ou aprendizagem de destilação, chamada assim por sua capacidade de ajudar um modelo mais complexo passar o seu aprendizado para um modelo mais simples; isto evita a perda de conhecimento prévio enquanto habilitando. Ele utiliza 2 modelos em conjunto, um modelo já treinado (professor), outro modelo sem treinamento (estudante). O resultado dos dois primeiros é utilizado para calcular a perda do estudante(Bergmann et al. 2020). Os parâmetros de optimização aqui será a função chamada de *AdamW* com uma *lr* de 0.00001 para ambos o professor e o estudante. A função de perda entre o professor e estudante é uma equação mais complexa com dois parâmetros arbitrários chamados de *alpha* (a) e temperatura (T). Primeiro é feito uma divisão entre as predições individuais com o parâmetro temperatura eles são subsequentemente processados por uma função chamada de *softmax*, $F(x) = exp(x)oversum(exp(x))$; após o

processamento ambas as perdas são colocadas na função de perda *KLDivergence*, como se as predições do professor (P_{pr}) fosse a verdade, e multiplicando o resultado pelo quadrado da temperatura, $D_p = (P_{pr} * LOG(P_{pr}/P_{es})) * (T^2)$. Com a equação de perda final sendo:

$$L_f = a * L_{es} + (1 - a) * D_{loss}$$

L_f é perda final e L_{es} é a perda normal do estudante. Será treinado uma versão com o detector Simples, pois a máquina utilizada para fazer este trabalho não tem VRAM suficiente para treinar o modelo *Resnet50* (He et al. 2015) junto com outro modelo.

Para a segunda variante era originalmente planejado fazer um modelo com um gerador de imagens para universalizar as imagens baseado na efetividade de *gans* em facilitar a detecção (Tran et al. 2021), mas com o hardware disponível não foi possível; principalmente, quando não é conhecido detalhes da estrutura interna do gerador para manter a paridade de testes. Portanto, foi explorado a utilização de algorítimos para a extração de informação da imagem e o efeito de diferentes algorítimos nos modelos. Quando foi utilizado o *resnet50* (He et al. 2015) como extrator, não foi achado nenhum ganho quando comparado ao uso do modelo completo. Além de não ter ganho, o aprendizado foi mais lento com a sua precisão não chegando a níveis altos. Houve um leve aumento na precisão médias com alguns modelos mas foi um aumento pequeno de 2% a 4%. Adicionar uma camada densa . Realço que, o extrator foi usado só para pegar informações pré-classificação. Para outros usos, como mapeamento da profundidade de RGB (Leporoni et al. 2024), foi obtido bons resultados.

	ADM	*SDV1.4*	*BigGan*	*MidJourney*	*Glide*	*VDQM*	*PM*
ProfessorC	62%	57%	91%	68%	89%	57%	71%
P&EC	64%	51%	94%	60%	92%	59%	70%
Resnet50C	55%	60%	93%	93%	90%	62%	76%
P&EAD	99%	50%	50%	50%	50%	50%	58%
P&ES1.4	54%	89%	48%	54%	52%	56%	59%
P&EBG	71%	49%	98%	51%	91%	66%	71%
P&EMI	63%	58%	70%	76%	79%	64%	68%
P&EGL	75%	49%	98%	53%	98%	68%	74%
P&EVD	93%	50%	89%	57%	92%	94%	79%

Tabela 4. Tabela com os teste de P&E. As letras maiúsculas no final denotam os dados usados durante o treinamento, como o professor sendo sempre o Professor treinado no conjunto. Resultados de testes com *Wukong* e *SDV1.5* foram removidos por motivos de formatação

Pelos resultados da tabela 4, pode-se ver que o modelo estudante obteve resultados similares a sua versão mais complexa quando treinado com o mesmo conjunto. Quando o estudante foi treinado com um modelo fora do conjunto, houve uma grande queda na precisão, principalmente com dados provenientes do *Midjourney*(Midjourney 2024) em que o detector já tinha problemas. Também foi observada uma melhoria na detecção para modelos que nunca foram usados em nenhuma parte do treinamento como os dados de *ADM* (Dhariwal and Nichol 2021) com o modelo *P&EVD*.

Por último, testes foram feitos para determinar a quantidade de dados necessários para uma classificação básica. Para isso, as imagens dos *datasets* originais.

	Era	Teste	Validação
Glide	1	55%	80%
Glide	2	65%	86%
Glide	3	71%	88%
Glide	4	75%	87%
Glide	5	77%	85%

Tabela 5. Versão Direta para o classificador

	Era	Teste	Validação
Glide2048	1	71%	61%
Glide2048	2	75%	91%
Glide2048	3	76%	91%
Glide2048	4	76%	89%
Glide2048	5	76%	81%

Tabela 6. Versão com uma camada extra de 2048 Neurônios

Começando com mil imagens por categoria e acrescentadas até resultados semelhantes ao conjunto completo, serem apresentados. Esse método não foi efetivo, o detector usando *ADM* (Dhariwal and Nichol 2021) para treinamento atingiu um resultado estável com apenas 5 mil imagens por categoria enquanto o detector que usou *BigGan* (Brock et al. 2019) para treinamento não atingiu acima de 60% em seu teste de validação, mesmo após a 20 mil imagens por categoria, mas, sempre houve um ganho nos testes de validação quando foi usado 5 mil imagens por categoria.

4. Conclusões, Recomendações e Observações

4.1. Recomendações Gerais

Baseado nos dados discutidos, a curto prazo, uma medida que poderia ser tomada contra *deepfakes* mal-intencionadas em redes-sociais, seria um estilo reacionário onde um vídeo que, ao atingir uma métrica ou ser reportado, é verificados por um detector treinado para conhecer os principais modelos e técnicas de geração de *deepfakes*. Mesmo que não seja um método perfeito, redes sociais já possuem filtros para conteúdos sensitivos e funções para reportar conteúdos que passem despercebidos por esses filtros. Só seria necessário dar a opção para os usuários insistirem na natureza falsa do conteúdo caso o detector secundário resulte em um negativo; este retorno, caso venha de uma quantidade significativa de usuários, seria a métrica usada para que uma pessoa faça este julgamento. Para médio a longo prazo, seria necessário um alto investimento para que os detectores sejam capazes de superarem os geradores, já que é necessário um esforço coletivo para estabelecer um conjunto de dados frescos necessário para o treinamento, e hardware capaz de processar está quantidade de dados com facilidade. Também seria efetivo a criação de um banco de dados que reuniria *datasets* de diferentes modelos para, centralizar mais os métodos e ferramentas. Experimentação com detectores com segmentação de objetos, merece ser explorado caso *deepfakes* cheguem ao seu extremo. Por último, recomendo a experimentação com a união de diversas formas de detecção aplicáveis.

4.2. Dificuldades Encontradas

Durante o trabalho houve uma dificuldade de achar *dataset* de *deepfakes* envolvendo humanos. As imagens fornecidas pela *generated photos* (Photos 2024) foi um bom começo, mas como foi discutido antes, utilizando somente uma fonte de dados para treinamento ou teste não é uma boa métrica ou prática para detecção de *deepfakes*, e conseguir mais dados vindos de outros geradores sem assistência é lento e ineficiente. Hardware, a placa de vídeo utilizada quebrou perto do começo do treinamento dos modelos, e o processo de descobrir que o problema era a placa e conseguir o dinheiro para comprar uma nova

atrasou o trabalho. Também houve problemas de compatibilidade quando a nova placa foi introduzida, algo que causou que os primeiros modelos levassem um tempo exagerado para serem treinados. Mesmo quando o problema foi resolvido e o detector Simples conseguia ser treinado em passo de 12 minutos por iteração, o detector utilizando o *resnet50* (He et al. 2015) na biblioteca *pytorch* (Paszke et al. 2019) ainda levava 1 hora por iteração, um número que dobrou quando os testes intermediários começaram. Este tempo não foi aceitável dado ao ambiente de trabalho, isto levou á um segundo treinamento do *resnet50* (He et al. 2015) no *Tensorflow* (Abadi et al. 2015). Algo que não foi frutífero para nada além da validação dos testes feitos no *pytorch* (Paszke et al. 2019), dado que a memória disponível não foi o suficiente para um treinamento P&E utilizando o *resnet50* (He et al. 2015) junto de outro modelo. No momento, a utilização de etiquetas não é propriamente utilizada devido as diferentes formas de se arquivá-las não serem compatíveis.

4.3. Pontos de referência Para Futuros Projetos de Detecção de *Deepfakes*

- Experimentação com a utilização de etiquetas e marcações denotando os objetos dentro da imagem, principalmente com o refinamento rápido da área de *deepfakes*
- Recomendo ter no mínimo 12GB de VRAM, mais de uma placa de vídeo e 24GB de RAM para um processamento de dados confortável.
- Verificação se há uma um teto para a criação e detecção de *deepfakes*
- Refinar as métricas de relação entre diferentes modelos geradores
- Arranjar uma forma rápida e eficiente para adquirir *deepfakes* de modelos mais recentes.
- Utilizar diferentes combinações de modelos gerativos para servirem como fontes de dados
- Conseguir no mínimo 5 mil imagens e as deixar bem separadas e rotuladas para uma análise melhor
- Recomendado a utilização de múltiplos tipos de análise

4.4. Conclusão Final

Em conclusão, foi olhada a história da detecção de *deepfakes*, viu-se que é um campo de pesquisa novo e altamente experimental que está sempre correndo atrás dos avanços de seu alvo. Foi visto o que é necessário para se criar um modelo gerativo de ponta; por conta disso pode-se inferir que um indivíduo, enquanto é capaz de criar modelos gerativos de boa qualidade, ainda não tem a capacidade de criar um modelo do calibre de grandes companhias, e não será capaz por um bom tempo. Foi explorado as particularidades de métodos de detecção de *deepfakes*. Por meio disso foi visto que, certos modelos são semelhantes em termos de detecção e que certos detectores conseguem perceber estas similaridades, dependendo da sua complexidade. Finalmente, foram feitas recomendações de metodologia e hardware para futuras tentativas de detecção de *deepfakes*.

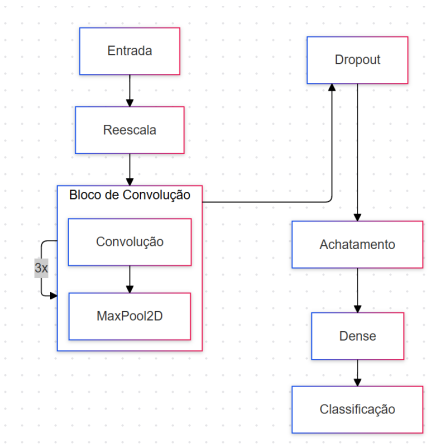

Figura 4. Diagrama visual do simples

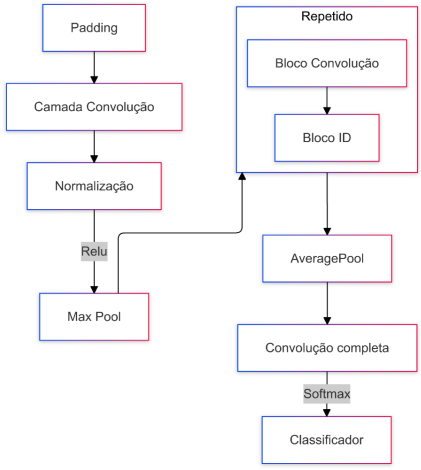

Figura 5. Diagrama visual do *resnet50* (He et al. 2015)

Referências

[Abadi et al. 2015] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., et al. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

[Arat 2018] Arat, M. M. (2018). Cross entropy for tensorflow.

[Bergmann et al. 2020] Bergmann, P., Fauser, M., Sattlegger, D., and Steger, C. (2020). Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

[Betker et al. 2024] Betker, J., Goh, G., Jing, L., Brooks, T., et al. (2024). Improving image generation with better captions.

[Bommasani et al. 2022] Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., et al. (2022). On the opportunities and risks of foundation models.

[Britton 2023] Britton, B. (2023). They appeared in deepfake porn videos without their consent. few laws protect them.

[Brock et al. 2019] Brock, A., Donahue, J., and Simonyan, K. (2019). Large scale gan training for high fidelity natural image synthesis.

[Chen et al. 2024] Chen, M., Mei, S., Fan, J., and Wang, M. (2024). An overview of diffusion models: Applications, guided generation, statistical rates and optimization.

[Cheng et al. 2022] Cheng, H., Guo, Y., Wang, T., Li, Q., et al. (2022). Voice-face homogeneity tells deepfake.

[Choi et al. 2018] Choi, Y., Choi, M., Kim, M., Ha, J.-W., et al. (2018). Stargan: Unified generative adversarial networks for multi-domain image-to-image translation.

[Dang et al. 2020] Dang, H., Liu, F., Stehouwer, J., Liu, X., and Jain, A. (2020). On the detection of digital face manipulation.

[Dhariwal and Nichol 2021] Dhariwal, P. and Nichol, A. (2021). Diffusion models beat gans on image synthesis.

[Dong et al. 2022] Dong, J., Wang, Y., Lai, J., and Xie, X. (2022). Restricted black-box adversarial attack against deepfake face swapping.

[Duffy 2024] Duffy, C. (2024). 'there are no guardrails.' this mom believes an ai chatbot is responsible for her son's suicide.

[Face 2024] Face, H. (2024). Text-to-image.

[Goodfellow et al. 2014] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., et al. (2014). Generative adversarial networks.

[Google 2023] Google (2023). Google's secure ai framework.

[Google 2024] Google (2024). Text-to-image ai.

[Gu et al. 2022] Gu, S., Chen, D., Bao, J., Wen, F., et al. (2022). Vector quantized diffusion model for text-to-image synthesis.

[He et al. 2015] He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition.

[Hukkelås et al. 2019] Hukkelås, H., Mester, R., and Lindseth, F. (2019). Deepprivacy: A generative adversarial network for face anonymization. In *Advances in Visual Computing*, pages 565–578. Springer International Publishing.

[Imagen-Team-Google et al. 2024] Imagen-Team-Google, :, Baldridge, J., Bauer, J., et al. (2024). Imagen 3.

[Intelligence 2024] Intelligence, M. (2024). Graphics processing unit (gpu) companies (2024 - 2029).

[Jia et al. 2022] Jia, S., Li, X., and Lyu, S. (2022). Model attribution of face-swap deepfake videos. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 2356–2360.

[Karras et al. 2018] Karras, T., Aila, T., Laine, S., and Lehtinen, J. (2018). Progressive growing of gans for improved quality, stability, and variation.

[Karras et al. 2019] Karras, T., Laine, S., and Aila, T. (2019). A style-based generator architecture for generative adversarial networks.

[Kong et al. 2020] Kong, J., Kim, J., and Bae, J. (2020). Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis.

[Leporoni et al. 2024] Leporoni, G., Maiano, L., Papa, L., and Amerini, I. (2024). A guided-based approach for deepfake detection: Rgb-depth integration via features fusion. *Pattern Recognition Letters*, 181:99–105.

[Li et al. 2024] Li, C., Huang, D., Lu, Z., Xiao, Y., et al. (2024). A survey on long video generation: Challenges, methods, and prospects.

[Lin et al. 2015] Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., et al. (2015). Microsoft coco: Common objects in context.

[Lindner 2024] Lindner, J. (2024). Average photo size demystified: From jpegs to raw and beyond.

[Midjourney 2024] Midjourney (2024). Midjourney site.

[Mori et al. 2012] Mori, M., MacDorman, K., and Kageki, N. (2012). The uncanny valley [from the field]. *IEEE Robotics & Automation Magazine*, 19:98–100.

[Naveed et al. 2024] Naveed, H., Khan, A. U., Qiu, S., Saqib, M., et al. (2024). A comprehensive overview of large language models.

[Nichol et al. 2022] Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., et al. (2022). Glide: Towards photorealistic image generation and editing with text-guided diffusion models.

[of Dublin 2024] of Dublin, U. C. (2024). Why do data analysts use python?

[Ojha et al. 2024] Ojha, U., Li, Y., and Lee, Y. J. (2024). Towards universal fake image detectors that generalize across generative models.

[Paszke et al. 2019] Paszke, A., Gross, S., Massa, F., Lerer, A., et al. (2019). Pytorch: An imperative style, high-performance deep learning library.

[Patel et al. 2023] Patel, M., Yuen, E. J., Stephenson, C., and Seguin, L. (2023). How we trained stable diffusion for less than $50k (part3)$.

[Photos 2024] Photos, G. (2024). Free dataset for academic research.

[Pytorch 2024] Pytorch (2024). Start locally.

[Radulovic 2018] Radulovic, P. (2018). Steam game pulled from store after allegations of cryptocurrency mining.

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

[Sacramento 2024] Sacramento, A. (2024). Deepfakes crescem 830

[Seymour et al. 2021] Seymour, M., Yuan, L., Dennis, A., and Riemer, K. (2021). Have we crossed the uncanny valley? understanding affinity, trustworthiness, and preference for realistic digital humans in immersive environments. *Journal of the Association for Information Systems*, 22:591–617.

[Systems 2024] Systems, P. (2024). Hardware recommendations for machine learning / ai.

[Team et al. 2024] Team, G., Anil, R., Borgeaud, S., Alayrac, J.-B., et al. (2024). Gemini: A family of highly capable multimodal models.

[Tran et al. 2021] Tran, N.-T., Tran, V.-H., Nguyen, N.-B., Nguyen, T.-K., and Cheung, N.-M. (2021). On data augmentation for gan training. *IEEE Transactions on Image Processing*, 30:1882–1897.

[Trindade and Oliveira 2024] Trindade, A. S. C. E. d. and Oliveira, H. P. C. d. (2024). InteligÊncia artificial (ia) generativa e competÊncia em informaÇÃo: Habilidades informacionais necessÁrias ao uso de ferramentas de ia generativa em demandas informacionais de natureza acadÊmica-cientÍfica. *Perspectivas em Ciência da Informação*, 29:e–47485.

[UFJF 2023] UFJF (2023). Mídias sociais e jornalismo: os perigos da desinformação.

[Walton 2023] Walton, J. (2023). Stable diffusion benchmarks: 45 nvidia, amd, and intel gpus compared.

[Wang and Su 2019] Wang, S. and Su, Z. (2019). Metamorphic testing for object detection systems.

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

[Wiener 2024] Wiener, Janet. Bronson, N. (2024). Facebook's top open data problems.

[Yang et al. 2021] Yang, Y., Liang, C., He, H., Cao, X., and Gong, N. Z. (2021). Faceguard: Proactive deepfake detection.

[Zhang et al. 2024] Zhang, D., Bohacek, M., and Kim, A. (2024). Awesome deepfakes detection.

[Zhang et al. 2022] Zhang, D., Lin, F., Hua, Y., Wang, P., et al. (2022). Deepfake video detection with spatiotemporal dropout transformer. In *Proceedings of the 30th ACM International Conference on Multimedia*, MM '22, page 5833–5841, New York, NY, USA. Association for Computing Machinery.

[Zhu et al. 2020] Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. (2020). Unpaired image-to-image translation using cycle-consistent adversarial networks.

[Zhu et al. 2023] Zhu, M., Chen, H., Huang, M., Li, W., et al. (2023). Gendet: Towards good generalizations for ai-generated image detection.