

LILIANA OLIVEIRA DA SILVA

**TRANSFERÊNCIA DE APRENDIZADO EM REDES NEURAIIS
CONVOLUCIONAIS PARA DETECÇÃO DE *DEEPPAKES* FACIAIS**

Trabalho de Conclusão de Curso apresentado
à banca avaliadora do Curso de Engenharia
de Computação, da Escola Superior de
Tecnologia, da Universidade do Estado do
Amazonas, como pré-requisito para obtenção
do título de Engenheiro de Computação.

Orientador(a): Profa. Dra. Elloá Barreto Guedes da Costa

Manaus – Junho – 2026

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).
Sistema Integrado de Bibliotecas da Universidade do Estado do Amazonas.

S586t	Silva, Liliana Oliveira da Transferência de Aprendizado em Redes Neurais Convolucionais para Detecção de Deepfakes Faciais / Liliana Oliveira da Silva. Manaus : [s.n], 2026. 50 f.: color.; 21.0 cm. TCC - Graduação em Engenharia de Computação- Universidade do Estado do Amazonas, Manaus, 2026. Orientador: Elloá Barreto Guedes da Costa. 1. Deepfakes. 2. Visão Computacional. 3. Redes Neurais Convolucionais. 4. Transferência de Aprendizado. 5. Inteligência Artificial. I. Elloá Barreto Guedes da Costa (Orient.) II. Universidade do Estado do Amazonas. III. Título CDU(1997)62:004
-------	---



FOLHA DE APROVAÇÃO

Transferência de Aprendizado em Redes Neurais Convolucionais para Detecção de Deepfakes Faciais

Liliana Oliveira da Silva

Trabalho de Conclusão de Curso I defendido e aprovado pela banca avaliadora
constituída pelos professores:

Profa. Dra. Elloá Barreto Guedes da Costa
Presidente

Prof. Dra. Áurea Hiléia da Silva Melo
Avaliador

Prof. Dr. Sergio Cleger Tamayo
Avaliador

Manaus, 23 de junho de 2025.

Resumo

A proliferação de mídias sintéticas altamente verossímeis (*deepfakes*) exige métodos robustos de detecção. Este trabalho investiga a eficácia de Redes Neurais Convolucionais (VGG16, ResNet-152 e Inception-v3) na identificação de *deepfakes* faciais, empregando Transferência de Aprendizado. A metodologia englobou experimentos de extração de características e ajuste fino (*fine-tuning*) em uma base de dados realista da literatura. Os resultados internos evidenciaram a arquitetura ResNet-152 com o melhor desempenho, alcançando acurácia superior a 99% após o ajuste fino. Contudo, testes de validação externa em imagens inéditas revelaram uma queda de desempenho das redes, destacando o crítico desafio de generalização dos modelos frente a novos métodos de síntese. Conclui-se que, embora as arquiteturas avaliadas sejam altamente eficazes no domínio conhecido, é imprescindível desenvolver estratégias que ampliem a capacidade de generalização das redes para acompanhar a rápida evolução das tecnologias de Inteligência Artificial Generativa.

Palavras-Chave: Deepfakes. Visão Computacional. Redes Neurais Convolucionais. Transferência de Aprendizado. Inteligência Artificial.

Abstract

The proliferation of highly realistic synthetic media (deepfakes) demands robust detection methods. This study investigates the effectiveness of Convolutional Neural Networks (VGG16, ResNet-152, and Inception-v3) in identifying facial deepfakes using Transfer Learning. The methodology encompassed feature extraction and fine-tuning experiments on a realistic dataset from the literature. Internal results highlighted the ResNet-152 architecture as the best-performing model, achieving over 99% accuracy after fine-tuning. However, external validation tests on unseen images revealed a drop in network performance, emphasizing the critical generalization challenge models face against novel synthesis methods. We conclude that while the evaluated architectures are highly effective within known domains, it is essential to develop strategies that enhance their generalization capacity to keep pace with the rapid evolution of generative artificial intelligence technologies.

Keywords: Deepfakes. Computer Vision. Convolutional Neural Networks. Transfer Learning. Artificial Intelligence.

Agradecimentos

Aos meus familiares, pelo apoio contínuo, incentivo e compreensão ao longo de toda a minha formação acadêmica e durante a realização deste Trabalho de Conclusão de Curso.

À minha orientadora, pela paciência, dedicação e atenção aos detalhes. Agradeço por compartilhar seus conhecimentos, por ter acreditado em mim e pelo apoio durante todo o tempo, guiando o desenvolvimento desta pesquisa com foco nos resultados, o que foi fundamental para concluirmos o trabalho considerando o tempo que tínhamos.

Aos meus amigos, especialmente Rafael, por todo o suporte oferecido ao longo da execução deste projeto.

Ao Laboratório de Sistemas Inteligentes, pela disponibilização da infraestrutura e dos recursos essenciais para a realização dos experimentos e desenvolvimento deste trabalho.

Agradeço também a todos que, direta ou indiretamente, contribuíram para a conclusão desta etapa.

Sumário

Lista de Tabelas	viii
Lista de Figuras	x
1 Introdução	1
1.1 Contextualização	1
1.2 Objetivos	3
1.3 Metodologia	3
1.4 Justificativa	4
1.5 Transparência da IA e Considerações Éticas	5
1.6 Organização do Documento	5
2 Fundamentação Teórica	6
2.1 IA, Aprendizado de Máquina e Aprendizado Profundo	6
2.2 Redes Neurais Artificiais	9
2.3 Transferência de Aprendizado	12
2.4 Redes Neurais Convolucionais	14
2.4.1 VGG-16	17
2.4.2 ResNet-152	18
2.4.3 Inception-v3	21
2.5 Trabalhos Relacionados	22
3 Material e Métodos	25

3.1	Dados Experimentais	25
3.2	Preparação das Amostras	27
3.3	Modelos, Parâmetros e Hiperparâmetros	29
3.3.1	Experimento 1	29
3.3.2	Experimento 2	30
3.4	Avaliação de Desempenho	30
3.5	Ambiente de Desenvolvimento	31
4	Resultados e Discussão	32
4.1	Resultados do Experimento 1	32
4.2	Resultados do Experimento 2	35
4.3	Análise Comparativa dos Experimentos	38
4.4	Validação Externa	40
5	Considerações Finais	45

Lista de Tabelas

2.1	Análise comparativa de métodos recentes de detecção de <i>deepfakes</i> , destacando os conjuntos de dados de treinamento e teste, a métrica de Melhor AUC (%) e a utilização de aprendizado de transferência.	24
3.1	Especificações de parâmetros e hiperparâmetros do Experimento 1 por modelo .	30
3.2	Especificações de parâmetros e hiperparâmetros do Experimento 2 por modelo .	30
4.1	Métricas de desempenho obtidas no conjunto de teste.	34
4.2	Métricas de desempenho obtidas no conjunto de teste no Experimento 2.	37
4.3	Métricas de desempenho obtidas no conjunto de teste nos Experimento 1 e 2 . .	39
4.4	Tempo de treinamento e total de parâmetros por experimento.	39
4.5	Distribuição absoluta de falsos negativos (erros) cometidos pelos modelos em função de subconjunto da classe sintetizada (<i>fake</i>)	40
4.6	Desempenho geral das arquiteturas na validação externa mediante <i>deepfakes</i> sintetizados via DiT-XL/2	42
4.7	Métricas detalhadas por classe na validação externa mediante <i>deepfakes</i> sintetizados via DiT-XL/2.	42
4.8	Desempenho geral das arquiteturas na validação externa mediante <i>deepfakes</i> sintetizados via StyleGAN3.	43
4.9	Métricas detalhadas por classe na validação externa mediante <i>deepfakes</i> sintetizados via StyleGAN3.	43

Lista de Figuras

2.1	Relação hierárquica entre IA, Aprendizado de Máquina e Aprendizado Profundo	8
2.2	Representação do modelo matemático de um neurônio artificial.	9
2.3	Exemplos de função de ativação	10
2.4	Esquema de mapeamento não linear de um conjunto de entradas $x_1, \dots, x_d$ para um conjunto de saída $y_1, \dots, y_c$	11
2.5	Arquitetura de uma rede neural <i>feedforward</i> MLP.	11
2.6	Diagrama da transferência de aprendizado.	13
2.7	Fine tuning	15
2.8	Duas etapas da operação de convolução. O filtro K 3×3 (azul) é multiplicado pela região de mesmo tamanho (laranja) da entrada I 7×7 e a soma do produto é usada para obter a entrada correspondente (roxo) no mapa de características. .	16
2.9	Progressão do aprendizado em uma rede neural convolucional. A rede aprende características a baixo nível nas camadas iniciais, seguido por características mais complexas nas camadas seguintes, que são usadas na tarefa de classificação. . .	17
2.10	Arquitetura da VGG16 com destaque para camadas de convolução (em azul), <i>pooling</i> (cinza) e <i>fully connected</i> (laranja)	19
2.11	Bloco residual usado no aprendizado residual profundo.	20
2.12	Estrutura do bloco Inception	21
2.13	Estatísticas de artigos publicados sobre geração e detecção de DeepFake. Dados obtidos da Web of Science, até dezembro de 2024	22
3.1	Distribuição de imagens do conjunto de dados por tipo	26

3.2	Distribuição média das dimensões das imagens. O ponto representa a média da largura e da altura, enquanto as barras indicam o desvio padrão em cada dimensão.	26
3.3	Imagens reais (primeira linha) e imagens sintetizadas (segunda linha).	27
3.4	Distribuição de imagens do conjunto de dados por partição	28
3.5	Distribuição de imagens do conjunto de dados por partição e tipo	28
3.6	Composição dos conjuntos de amostras de treinamento, teste e validação	29
3.7	Contagem do número de imagens segundo partição e classe	29
4.1	Curvas de perda (à esquerda) e acurácia (à direita) de treinamento e validação para o Experimento 1 (<i>feature extraction</i>).	33
4.2	Matrizes de confusão obtidas no conjunto de teste para o Experimento 1 (<i>feature extraction</i>).	34
4.3	Exemplos visuais da classe sintetizada classificados corretamente de forma exclusiva por ResNet-152 no Experimento 1.	35
4.4	Curvas de perda (à esquerda) e acurácia (à direita) de treinamento e validação para o Experimento 2 (<i>fine-tuning</i>).	36
4.5	Matrizes de confusão obtidas no conjunto de teste para o Experimento 2 (<i>fine-tuning</i>).	37
4.6	Colagem de imagens classificadas incorretamente como reais por todas as arquiteturas avaliadas no Experimento 2	38
4.7	Comparativo do custo temporal dos experimentos	40
4.8	Distribuição de erros por subconjuntos da classe sintetizada para o modelo ResNet-152 por experimento.	41
4.9	Acurácia na validação externa mediante <i>deepfakes</i> sintetizados via DiT-XL/2	42
4.10	Acurácia na validação externa mediante <i>deepfakes</i> sintetizados via StyleGAN3	44
4.11	Revocação por modelo na validação externa por modelo e conjunto	44

Capítulo 1

Introdução

1.1 Contextualização

É de conhecimento geral que o ritmo rápido do desenvolvimento da Inteligência Artificial (IA) e o lançamento público de ferramentas como ChatGPT, GitHub Copilot e DALL-E atraíram ampla atenção e otimismo pela comunidade em geral, mas também alarme por parte de pesquisadores, órgão reguladores, juristas e outros, devido a preocupações urgentes com a segurança dos dados, a disseminação automatizada de desinformação, os impactos éticos e outros (BRYNJOLFSSON; LI; RAYMOND, 2025). Tais ferramentas são exemplos de IA generativa, uma classe de tecnologias de Aprendizado de Máquina capaz de gerar novos conteúdos, como textos, imagens, músicas e vídeos ao analisar padrões em dados existentes (AYDIN; KARAARSLAN, 2023).

Sob a perspectiva de Jaidka et al. (2025), a IA Generativa tornou a criação de conteúdo acessível a um público mais amplo por meio do uso de interfaces que exigem pouca habilidade técnica. As vantagens da IA Generativa se estendem por vários domínios, permitindo que os usuários gerem conteúdo envolvente, sintetizem informações de diversas fontes, traduzam dados complexos em resumos compreensíveis, dentre outras tarefas.

Apesar das vantagens, a dependência da IA Generativa para a criação de conteúdo também tem suas armadilhas. Na visão de Park e Nan (2025) a fluência, escala e adaptabilidade dos modelos de IA generativa oferecem oportunidades e riscos. Esses últimos não tem precedentes

para a forma como a informação é gerada, consumida e confiada. Tais autores reiteram que a IA pode ser explorada intencionalmente por agentes mal-intencionados para fabricar desinformação convincente, mídias sintéticas ou até mesmo relatórios científicos falsificados (PARK; NAN, 2025).

Os avanços em IA Generativa e a consequente transformação na criação de conteúdos digitais impulsionou o surgimento dos *deepfakes*. Mirsky e Lee (2021) descreve a palavra *deepfake* como uma combinação dos termos em língua inglesa “*Deep Learning*” e “*Fake*” e define o seu uso para denominar vídeos ou imagens gerados por modelos de IA Generativa, em que o rosto de uma pessoa é substituído pelo de outra. Atualmente o termo é usado de forma mais abrangente, em que *deepfakes* são representações sintéticas altamente verossímeis de pessoas, visando mimetizar a realidade de forma indistinguível para observadores humanos. Embora esse tipo de conteúdo possa ser empregado de maneira positiva, há várias formas de utilização ilegal ou maliciosa, tais como a produção de pornografia não consensual e a disseminação de desinformação (RANA et al., 2022).

Há uma preocupação em nível global com a ameaça dos *deepfakes* para desinformação. A UNESCO, por exemplo, afirma que os *deepfakes* diferem fundamentalmente da desinformação tradicional pois são convincentes, escaláveis e cada vez mais acessíveis. Os *deepfakes* não apenas introduzem falsidades em nosso ecossistema de informação mas também corroem os próprios mecanismos pelos quais as sociedades constroem um entendimento compartilhado (NAFFI, 2025). A ameaça da desinformação, intencional ou não, agora massivamente amplificada por mídias sintéticas, é classificada pelo Fórum Econômico Mundial como um dos principais riscos globais, destacando a linha cada vez mais tênue entre conteúdos gerados por IA e por humanos. A proliferação dessas mídias sintéticas representa uma ameaça sem precedentes à segurança digital, à privacidade individual e à confiabilidade da informação pública, podendo ser utilizada para fraudes financeiras, difamação e manipulação política em larga escala (World Economic Forum, 2025).

Em face do que foi exposto, o presente trabalho propõe a investigação da utilidade dos modelos de *Deep Learning* mediante Transferência de Aprendizado no combate aos *deepfakes*

faciais, contribuindo assim para mitigar os desafios enfrentados no contexto atual decorrentes do uso malicioso das técnicas de IA Generativa.

1.2 Objetivos

O objetivo geral deste trabalho consiste em avaliar a eficácia e eficiência da Transferência de Aprendizado em Redes Neurais Convolucionais Profundas na tarefa de detecção de *deepfakes* faciais.

1. Selecionar e preparar uma base de dados da literatura representativa do problema em questão;
2. Parametrizar, treinar e testar modelos de redes neurais convolucionais da literatura mediante transferência de aprendizado;
3. Analisar comparativamente os resultados obtidos.

1.3 Metodologia

Para atingir os objetivos propostos no escopo deste trabalho, a condução das atividades obedeceu à metodologia apresentada a seguir, composta dos seguintes passos:

1. Estudo dos conceitos relacionados à *Deep Learning*, Redes Neurais Convolucionais e Transferência de Aprendizado;
2. Estudo do ferramental tecnológico para implementação e aferição de desempenho de Redes Neurais Convolucionais, incluindo Pytorch, sci-kit learn, matplotlib, etc.;
3. Pesquisar uma base de dados representativa de *deepfakes* faciais;
4. Descrever o problema como uma tarefa de classificação binária, estabelecendo os atributos, forma de avaliação e métricas de desempenho;
5. Efetuar a limpeza e preparação da base de dados;

6. Elencar diferentes modelos pré-treinados para a tarefa considerada;
7. Treinar os modelos com os exemplos da base de dados;
8. Testar os modelos e coletar métricas de desempenho;
9. Analisar os resultados obtidos identificando os modelos mais adequados ao cenário considerado.

1.4 Justificativa

Lançado durante a 5ª Conferência Nacional de Ciência, Tecnologia e Inovação, o Plano Brasileiro de Inteligência Artificial (PBIA) (BRASIL. Ministério da Ciência, Tecnologia e Inovação, 2024) é a principal política pública nacional para o desenvolvimento e uso responsável da IA no Brasil. O plano busca desenvolver soluções em IA que melhorem significativamente a qualidade de vida da população, otimizando a entrega de serviços públicos e promovendo a inclusão social. No documento publicado pelo governo brasileiro o uso da IA é contextualizado como segue:

“Primeiramente, a IA tem o potencial de transformar profundamente a democracia e a integridade da informação. Ela pode tanto fortalecer processos democráticos por meio de análises mais precisas de dados que ajudam a aperfeiçoar a entrega de serviços e bens públicos, quanto representar riscos à democracia, exigindo vigilância constante contra a disseminação de desinformação que afeta a percepção dos cidadãos sobre a realidade social.” (BRASIL. Ministério da Ciência, Tecnologia e Inovação, 2024, p. 17)

A preocupação mencionada no PBIA é corroborada por pesquisas recentes que mostram também que, em razão da maior exposição *online*, populações com baixo nível de letramento digital, como a população brasileira, mostram-se vulneráveis frente ao avanço da circulação de conteúdo sintético, com dificuldade em identificar a veracidade de conteúdo gerado por modelos de IA generativa que simulam uma situação ou realidade (SANCHES; MENEZES, 2025). Considerando essa demanda nacional, o presente trabalho, focado no desenvolvimento de modelos para

detectar *deepfakes* faciais, atua de forma convergente com o PBIA, fomentando seus objetivos de soberania e segurança digital.

Dado o caráter precursor dos métodos e técnicas de *Deep Learning* considerados, este trabalho propõe analisar a adequação dessas soluções a um problema prático com dados reais, contrastando suas vantagens e limitações com o estado da arte. Para a futura bacharel em Engenharia da Computação, a proposta viabiliza a prática de conceitos e ferramentas de uma área emergente. Adicionalmente, o trabalho possui relevância institucional, pois visa colaborar com as atividades científicas do Laboratório de Sistemas Inteligentes, vinculado ao Grupo de Pesquisas em Sistemas Inteligentes da Escola Superior de Tecnologia (EST/UEA).

1.5 Transparência da IA e Considerações Éticas

A ferramenta Google Gemini 3.1 Pro foi utilizada para revisão ortográfica do texto, o qual foi novamente revisado pela autora e orientadora. Ademais, essa ferramenta foi utilizada para aprimoramento dos gráficos e figuras ilustrativos do trabalho, tais como nas tarefas de rotulação de eixos, espaçamentos, colorações, posicionamento de textos e ajustes finos visuais, sempre com supervisão final da autora do trabalho.

1.6 Organização do Documento

Para melhor compreensão, este trabalho está estruturado em cinco capítulos, em que este primeiro apresentou uma sucinta introdução ao tema. Em seguida, o Capítulo 2 aborda os fundamentos teóricos necessários a execução do projeto, incluindo Inteligência Artificial, Redes Neurais Convolucionais e Transferência de Aprendizado. O Capítulo 3 detalha a metodologia empregada para detecção de *deepfakes*, descrevendo os dados utilizados, experimentos propostos e critérios de aferição de desempenho. A apresentação e análise dos resultados obtidos ocorrem no Capítulo 4 e, por fim, as considerações finais encontram-se no Capítulo 5.

Capítulo 2

Fundamentação Teórica

2.1 IA, Aprendizado de Máquina e Aprendizado Profundo

Diversas definições para IA (IA) se tornaram conhecidas ao longo do tempo. Uma definição de IA bastante aceita, apresentada em por John McCarthy em um *workshop* de verão em 1956 diz que a IA consiste em permitir que uma máquina simule o comportamento inteligente dos humanos com a maior precisão possível. Em 1994, Rich, Knight et al. (1994) propõem um novo conceito: IA é o estudo de como fazer computadores realizarem tarefas que, até o momento, os seres humanos fazem melhor. Apesar de os autores ressaltarem a efemeridade desse ponto de vista, essa noção se tornou obsoleta à medida que a IA se tornou mais robusta e complexa, capaz de resolver um número significativo de tarefas. Mais perto dos dias atuais, Chollet (2021) oferece uma versão breve na qual a IA é caracterizada como o esforço para automatizar tarefas intelectuais normalmente realizadas por humanos. Essa interpretação ganha um bom complemento no texto anterior de Russell e Norvig (2016): a disciplina de IA se esforça para construir entidades inteligentes bem como compreendê-las.

Apesar de nos anos iniciais a IA provar ter capacidade de resolver problemas lógicos conhecidos, como jogar xadrez, se tornou resistente quando confrontada a problemas complexos como classificação de imagens, reconhecimento de voz e tradução. E então, para resolver tais

adversidades, um subcampo de estudo ganhou posição de destaque: o Aprendizado de Máquina. Para Sarkar, Bali e Ghosh (2018), a definição de Mitchell (1997) capta a essência do que é Aprendizado de Máquina de uma forma muito concisa. Segue

Um programa de computador é dito aprender a partir da experiência E , com relação a alguma tarefa T e alguma medida de desempenho P , se o seu desempenho em T , conforme medido por P , melhora com a experiência E . (MITCHELL, 1997, p. 2)

Nessa definição relativamente formal da palavra “tarefa”, o processo de aprendizagem em si não é a tarefa. Aprender é o meio de adquirir a capacidade de realizar a tarefa. O Aprendizado de Máquina então permite resolver tarefas que são demasiadamente difíceis de solucionar usando programas escritos e pensados por humanos (GOODFELLOW et al., 2016).

Chollet (2021) e Khan et al. (2018) reiteram que o um programa de Aprendizado de Máquina é treinado em vez de explicitamente programado. O programa é apresentado a vários exemplos, observações do mundo real formando um conjunto de treinamento e, a partir desses dados, encontra um caminho, usando matemática, a fim de resolver a tarefa para o qual foi designado. Goodfellow et al. (2016) usa esse caminho para descrever Aprendizado de Máquina quando diz que o campo é essencialmente uma forma de estatística aplicada com maior ênfase no uso de computadores para estimar funções estatisticamente complexas.

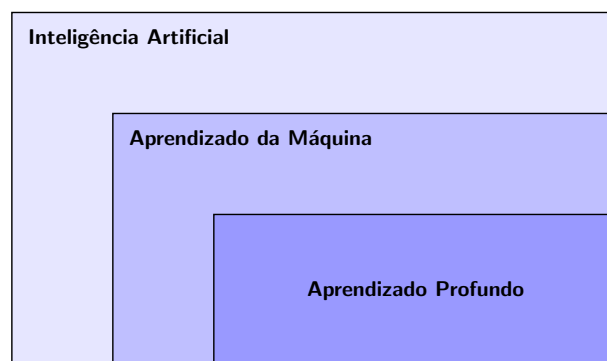
Aprendizado supervisionado e não supervisionado são as principais categorias nas quais os algoritmos de Aprendizado de Máquina podem ser divididos. As duas categorias podem ser diferenciadas pelo tipo de experiência que o algoritmo é permitido ter durante o processo de aprendizado (GOODFELLOW et al., 2016). No aprendizado supervisionado, a experiência E (dados de treinamento) constitui uma coleção de pares (dado: x , classe: y), cujo objetivo é estimar uma previsão y^* a partir de um exemplo x^* . No aprendizado não supervisionado, por sua vez, utilizam-se apenas os dados de entrada X diante da ausência de classes y , o que denota a inexistência de saídas verdadeiras para o mapeamento (JORDAN; MITCHELL, 2015).

Apesar dos algoritmos de Aprendizado de Máquina serem suficientes para resolver variados problemas, esses problemas se tornam muito custosos quando o número de dimensões dos

dados é alto. Esses desafios motivaram o desenvolvimento de um outro campo de pesquisa: Aprendizado Profundo (*Deep Learning*).

No campo do Aprendizado de Máquina, busca-se descobrir automaticamente regras capazes de mapear dados de entrada para uma saída desejada. Nesse processo, a criação de representações apropriadas dos dados de entrada desempenha um papel fundamental (SARKAR; BALI; GHOSH, 2018). Sendo assim, o problema central em Aprendizado de Máquina e Aprendizado Profundo consiste em transformar os dados de entrada de forma significativa, gerando representações que se aproximem da saída esperada (CHOLLET, 2021). Sob essa perspectiva, Chollet (2021) define o Aprendizado Profundo como um subcampo do Aprendizado de Máquina que introduz uma nova abordagem para extrair tais representações, enfatizando o aprendizado incremental por meio de camadas sucessivas. A partir dessa definição, infere-se que o termo profundo refere-se justamente à quantidade de camadas consecutivas que constituem a arquitetura do modelo. A relação hierárquica entre os conceitos apresentados é ilustrada na Figura 2.1.

Figura 2.1: Relação hierárquica entre IA, Aprendizado de Máquina e Aprendizado Profundo



Fonte: Adaptado de CHOLLET (2021, p. 4).

Sistemas de Aprendizado Profundo utilizam algoritmos de otimização baseados em gradiente para ajustar parâmetros em toda essa rede neural multicamadas com base em erros em sua saída. Explorando arquiteturas modernas de computação paralela, como unidades de processamento gráfico (GPUs) originalmente desenvolvidas para jogos eletrônicos, foi possível construir sistemas de Aprendizado Profundo que contêm bilhões de parâmetros e que podem ser treinados

nas vastas coleções de imagens, vídeos e amostras de fala disponíveis na internet (JORDAN; MITCHELL, 2015).

2.2 Redes Neurais Artificiais

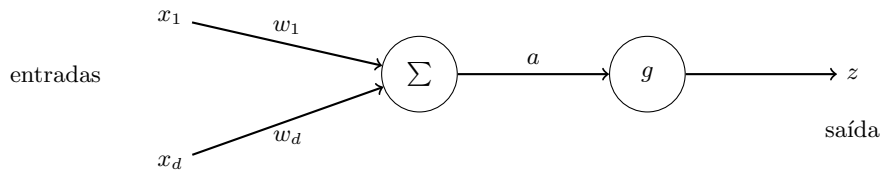
Um rede neural é, em essência, um processo abstrato de processamento de dados, composto por unidades de processamento simples denominadas neurônios. Esses neurônios apresentam uma notável aptidão para armazenar conhecimento adquirido por meio de experiências e disponibilizá-lo conforme necessário (HAYKIN, 2009). O termo *neurônio* é usado nesse contexto pois redes neurais são inspiradas no funcionamento do córtex cerebral de mamíferos. É necessário notar que, apesar do uso da palavra, esses modelos não aproximam estreitamente o funcionamento, a escala e a complexidade do cérebro humano (KHAN et al., 2018).

Um simples modelo matemático do neurônio foi introduzido por McCulloch e Pitts (1943) e tem a forma apresentada na Figura 2.2. O modelo de McCulloch-Pitts de um único neurônio forma uma soma ponderada das entradas $x_1, \dots, x_d$, dada por

$$a = \sum_{i=1}^d w_i x_i, \quad (2.1)$$

e então transforma essa soma utilizando uma função de ativação não linear $g(\cdot)$ para produzir a saída final $z = g(a)$ (BISHOP, 1995).

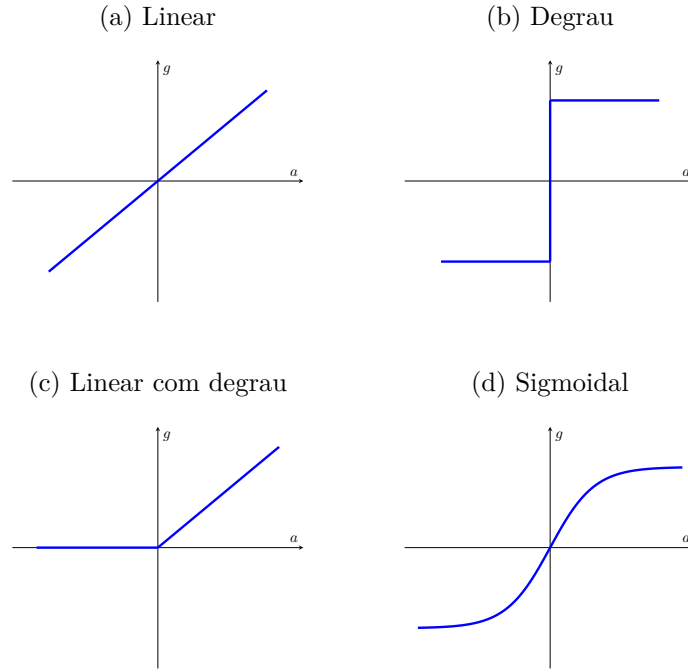
Figura 2.2: Representação do modelo matemático de um neurônio artificial.



Para o neurônio de McCulloch-Pitts, a função de ativação não linear é do tipo degrau deslocada do limiar de ativação θ em relação à origem, ou seja, a saída z será 1 para $\sum_{i=1}^d w_i x_i \geq \theta$ e 0 para $\sum_{i=1}^d w_i x_i < \theta$ (BRAGA; LUDERMIR; CARVALHO, 2000).

Bishop (1994) destaca alguns exemplos de função de ativação que são vistos na Figura x, incluindo o tipo degrau deslocada do limiar de ativação Figura 2.3b.

Figura 2.3: Exemplos de função de ativação

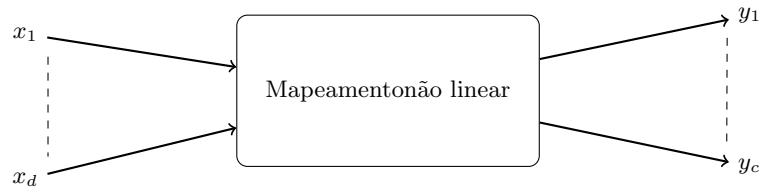


Fonte: Adaptado de BISHOP (1994, p. 1805).

Redes neurais artificiais podem ser agrupadas em dois grupos principais segundo a maneira como a informação se propaga na rede: redes recorrentes e redes alimentadas para a frente (*feedforward*) (KHAN et al., 2018). Segundo Bishop (1995), uma rede *feedforward* é basicamente uma função não linear que converte um conjunto de entradas independente $x_1, \dots, x_d$ em um conjunto de saídas $y_1, \dots, y_c$, conforme Figura 2.4. Nesse tipo de modelo, os neurônios são organizados em camadas, a entrada é transmitida em apenas uma direção e não há conexões recorrentes. Exemplos desse grupo são redes neurais convolucionais e redes Perceptron de Múltiplas Camadas (MLPs – *Multilayer Perceptron*).

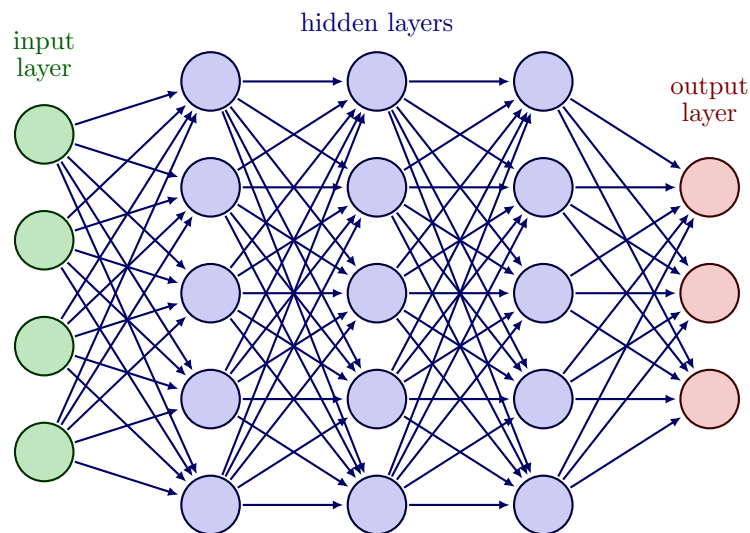
A Figura 2.5 ilustra um exemplo da arquitetura de uma rede MLP. A partir do diagrama, observa-se a organização estrutural em camadas que caracteriza uma rede neural. Cada uma delas é composta por um número n de neurônios, também denominados unidades de processamento. O fluxo de informações inicia-se na camada de entrada, onde os dados são recebidos, e propaga-se até a camada de saída, responsável por gerar as previsões finais. As camadas inter-

Figura 2.4: Esquema de mapeamento não linear de um conjunto de entradas $x_1, \dots, x_d$ para um conjunto de saída $y_1, \dots, y_c$.



mediárias, encarregadas do processamento e da extração de características, são denominadas camadas ocultas (KHAN et al., 2018).

Figura 2.5: Arquitetura de uma rede neural *feedforward* MLP.



Ao interligar vários neurônios é possível construir uma classe muito geral de mapeamentos não lineares, que pode ser aplicada a uma ampla gama de problemas práticos. A adaptação dos valores dos pesos, de acordo com um algoritmo de treinamento apropriado, permite que as redes aprendam em resposta a dados externos (BISHOP, 1995). Os pesos precisam ser ajustados devidamente para que a saída esperada possa ser alcançada a partir da rede neural e desse modo configuram o “modelo” gerado com base no conjunto de treinamento. Uma vez que o número de pesos é elevado, o processo de ajuste precisa ser automático (KHAN et al., 2018).

Segundo Braga, Ludermir e Carvalho (2000), o algoritmo de treinamento de redes MLP mais

popular é o *back-propagation*. Esse algoritmo foi proposto por Rumelhart, Hinton e Williams (1986) e é baseado na regra delta de Widrow e Hoff (1988), que atualiza os pesos da rede com base na diferença entre a saída desejada e a saída prevista. O treinamento ocorre em duas fases: a primeira, chamada *forward*, é usada para definir uma saída $\hat{y}$ dada uma entrada x seguida pela fase *backwards*, que utiliza a saída desejada e saída prevista pela rede para atualizar os pesos de suas conexões (BRAGA; LUDERMIR; CARVALHO, 2000).

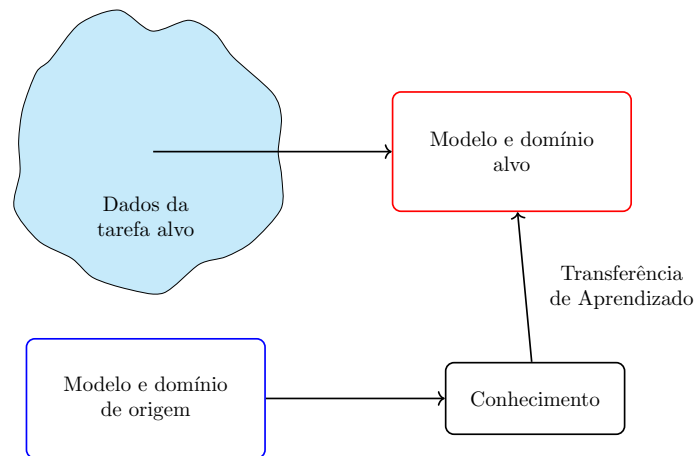
Goodfellow et al. (2016) explica o conceito da correção de erros usando um custo escalar $J(\theta)$ que é propagado pela rede usando o algoritmo de *back-propagation* a fim de calcular um gradiente e assim otimizar os pesos e ainda salienta, por outro lado, que *back-propagation* refere-se apenas ao método empregado para calcular o gradiente, enquanto outro algoritmo, como o gradiente descendente estocástico, é usado para realizar o aprendizado usando esse gradiente.

2.3 Transferência de Aprendizado

O trabalho de Pan e Yang (2009) destaca o elevado custo ou mesmo a impossibilidade de se coletar e rotular volumes significativos de dados em certos domínios, além da complexidade de treinar modelos do zero para tarefas específicas de Aprendizado de Máquina. Para contornar tais limitações, defende-se o uso de técnicas de Transferência de Aprendizado (Transfer Learning), cuja motivação é endossada de maneira similar por Sarkar, Bali e Ghosh (2018). Além da vantagem de reaproveitar o conhecimento de redes preexistentes e reutilizar modelos já consolidados, essa abordagem permite aplicar de forma inteligente o aprendizado prévio para solucionar novos problemas com maior celeridade ou eficácia.

A definição do conceito é bem sintetizada por Goodfellow et al. (2016): a transferência de aprendizado é a situação onde o que foi aprendido em um contexto é explorado para melhorar a generalização em um outro. Pan e Yang (2009) discrimina o processo em mais detalhes: dado um domínio de origem D_S e uma tarefa de aprendizado T_S , um domínio alvo D_T e uma tarefa de aprendizado T_T , o aprendizado por transferência tem como objetivo melhorar o aprendizado da função preditiva alvo $f_T(\cdot)$ em D_T utilizando o conhecimento em D_S e T_S , onde $D_S \neq D_T$ ou $T_S \neq T_T$. A Figura 2.6 complementa o entendimento dos conceitos em questão.

Figura 2.6: Diagrama da transferência de aprendizado.



Fonte: Adaptado de SARKAR; BALI; GHOSH (2018, p. 284).

Enquanto as técnicas tradicionais de Aprendizado de Máquina procuram aprender cada tarefa do zero, as técnicas de Transferência de Aprendizado buscam transferir o conhecimento de algumas tarefas anteriores para uma tarefa alvo (PAN; YANG, 2009). Goodfellow et al. (2016) confirma essa afirmação, mas salienta que a busca pode ser alcançada quando existem características úteis nos diferentes contextos, correspondentes a fatores subjacentes que aparecem em mais de um contexto. Isso é exemplificado por Zhang et al. (2023) ao mencionar modelos treinados no conjunto de dados ImageNet. Embora as imagens no ImageNet sejam em sua maioria não relacionadas uma certa categoria, os modelos nele treinados podem ser capazes de extrair características mais gerais de imagens que ajudam a identificar bordas, texturas, formas e composição de objetos, auxiliando em problemas gerais de classificação de imagens.

Pan e Yang (2009) classifica o método em três categorias diferenciadas pelos domínios e tarefas de origem e alvo. No aprendizado por transferência indutivo, a tarefa alvo é diferente da tarefa de origem, independentemente de os domínios de origem e alvo serem iguais ou não. Já no aprendizado por transferência transdutivo, as tarefas de origem e alvo são as mesmas, mas os domínios de origem e alvo são diferentes. Por último, o aprendizado por transferência não supervisionado foca na resolução de tarefas de aprendizado não supervisionado no domínio alvo, como agrupamento, redução de dimensionalidade e estimação de densidade.

Uma técnica comum de transferência de aprendizado é desenvolvida por Zhang et al. (2023)

e ilustrada na Figura 2.7. O *Fine tuning* (ajuste fino) compreende quatro passos principais:

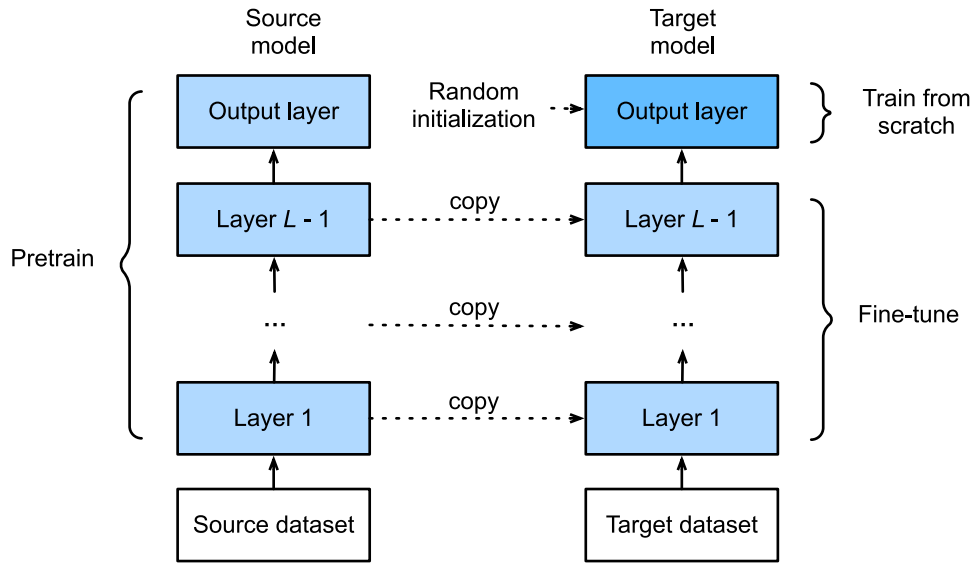
1. Pré-treinar um modelo em um conjunto de treinamento de origem, ImageNet por exemplo;
2. Criar um novo modelo alvo onde todo o desenho e parâmetros do modelo de origem são replicados, exceto a camada de saída;
3. Adicionar uma camada de saída ao modelo alvo, cujo tamanho de saída corresponda ao número de categorias do novo conjunto de dados e inicializar os parâmetros dessa camada de forma aleatória;
4. Treinar o modelo alvo em um conjunto de dados específico (por exemplo, um conjunto de dados de cadeiras).

A camada de saída será treinada do zero, enquanto os parâmetros de todas as camadas restantes serão ajustados (*fine-tuned*) com base nos parâmetros oriundos do modelo de origem. No passo 2, mencionado anteriormente, algumas condições são reconhecidas. Os parâmetros do modelo de origem contêm o conhecimento extraído do conjunto de dados de origem. Segundo, esse conhecimento será igualmente aplicável ao novo conjunto de dados alvo. Por fim, a camada de saída do modelo de origem está estritamente ligada aos rótulos do conjunto de dados origem e por isso não é utilizada no novo modelo. Assim sendo, neste exemplo, temos um exemplo de aprendizado por transferência indutivo.

2.4 Redes Neurais Convolucionais

Para Goodfellow et al. (2016), Redes Neurais Convolucionais (CNNs, do inglês, *Convolutional Neural Networks*) são redes neurais que usam convolução no lugar de um produto de matrizes em pelo menos uma de suas camadas. Convolução é um tipo especializado de operador linear que toma como entrada duas funções. No contexto de redes convolucionais, o primeiro argumento é referenciado, muitas vezes, como entrada e o segundo como filtro. A saída é chamada mapa de características.

Figura 2.7: Fine tuning

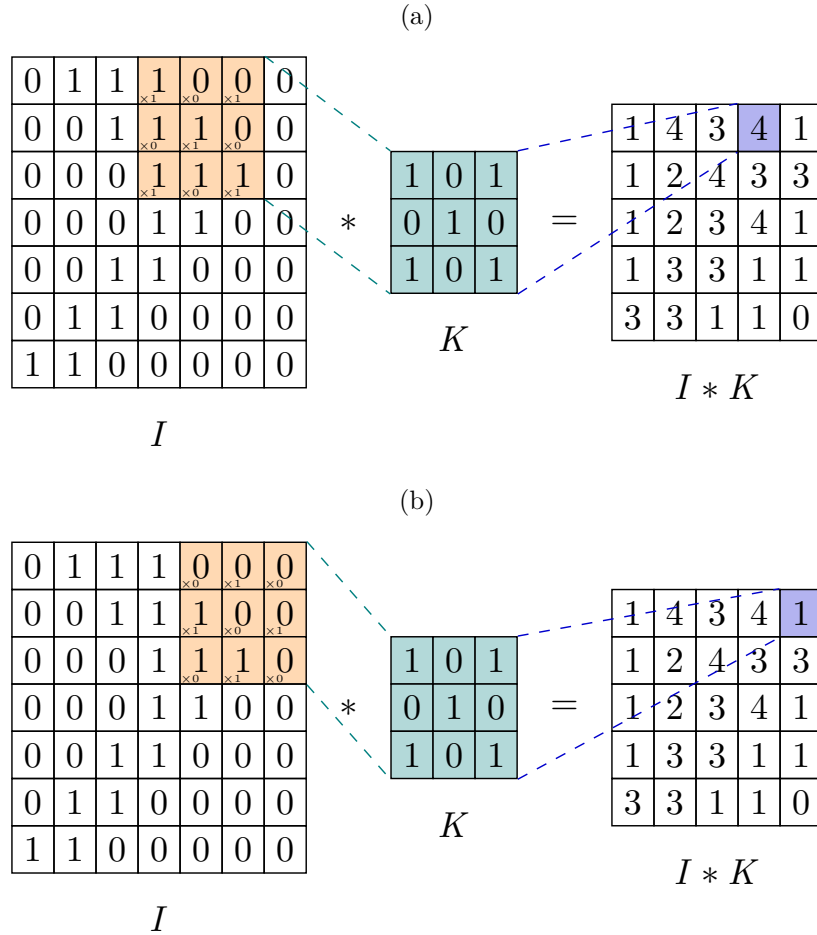


Fonte: ZHANG et al. (2023, p. 575)

Um exemplo de parte da operação de convolução, denotada pelo $*$, é visto nas Figuras 2.8a e 2.8b. A entrada I é uma matriz de dimensão 7×7 e o filtro convolucional K uma matriz 3×3 . A parte destacada na entrada é multiplicada pelo filtro e a soma do produto se torna um valor na saída, o mapa de características. O filtro continua a percorrer a extensão da entrada, nas duas dimensões, até que não haja mais dados a serem mapeados. No exemplo demonstrado, para calcular cada valor do mapa de características, o filtro dá um passo de 1 ao longo das posições horizontal e vertical. Este passo é denominado *stride* do filtro de convolução (KHAN et al., 2018). Outro ponto a se notar é a diferença entre as dimensões da entrada e do mapa de características. Para obter um mapa de características com as mesmas dimensões da entrada, *padding* pode ser usado. O *padding* consiste em adicionar um número apropriado de linhas e colunas em cada lado do mapa de características de entrada, de modo a tornar possível ajustar janelas de convolução centradas em cada elemento da entrada (CHOLLET, 2021).

Redes neurais convolucionais se encaixam entre as principais categorias de redes neurais quando deseja-se lidar com dados multidimensionais. Isso se deve ao fato de que cada unidade em uma camada na CNN é um filtro de duas ou n dimensões que realiza a operação de convolução com a entrada da camada. Isso é fundamental para os casos quando queremos aprender

Figura 2.8: Duas etapas da operação de convolução. O filtro K 3×3 (azul) é multiplicado pela região de mesmo tamanho (laranja) da entrada I 7×7 e a soma do produto é usada para obter a entrada correspondente (roxo) no mapa de características.

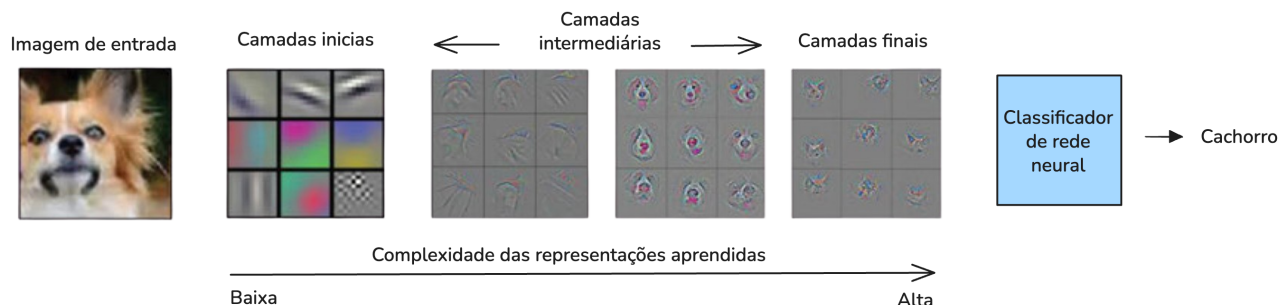


sobre dados multidimensionais, tal qual imagens e vídeos (KHAN et al., 2018).

As CNNs são úteis para aprendizado supervisionado e não supervisionado. Um exemplo de tarefa de classificação considerando aprendizado supervisionado é mostrado na Figura 2.1. A CNN aprende a mapear uma determinada imagem para uma categoria, detectando uma série de representações abstratas de características que variam de simples a mais complexas. Essas características distintas são então usadas dentro da rede para prever a categoria correta da imagem de entrada.

Um momento essencial para o endereçamento de CNNs a tarefas de Visão Computacional foi quando o seu uso viabilizou soluções para o *ImageNet Large-Scale Visual Recognition Challenge* (ILSVRC) (RUSSAKOVSKY et al., 2015). O desafio era considerado difícil para a época, no

Figura 2.9: Progressão do aprendizado em uma rede neural convolucional. A rede aprende características a baixo nível nas camadas iniciais, seguido por características mais complexas nas camadas seguintes, que são usadas na tarefa de classificação.



Fonte: Adaptado de KHAN et al. (2022, p. 42).

qual imagens precisavam ser classificadas em 1000 categorias distintas, após um treinamento com 1.4 milhões de imagens. Então, em 2012, Krizhevsky, Sutskever e Hinton (2012) alcançou uma acurácia de 83.6% com a CNN AlexNet e isso representou um avanço expressivo no estado da arte de classificação massiva de imagens. Desde então, as soluções propostas para o desafio passaram a ser dominadas por redes neurais convolucionais. Em 2015, o vencedor alcançou uma acurácia de 96.4% e o desafio foi tido como resolvido (CHOLLET, 2021). Como legado do ILSVRC, várias arquiteturas de redes neurais convolucionais para tarefas de classificação foram introduzidas, dentre as quais a VGG16, ResNet e Inception-v3 serão detalhadas a seguir.

2.4.1 VGG-16

A CNN VGG, acrônimo de *Visual Geometry Group* em inglês, é um dos modelos mais populares de redes neurais convolucionais, devido à sua simplicidade e o uso de filtros de dimensão baixa, resultando em redes profundas (KHAN et al., 2018), ou seja, comporta por várias camadas convolucionais sucessivas.

Simonyan e Zisserman (2014) propuseram um conjunto de configurações de rede. Cada elemento nesse conjunto faz uso de blocos. Zhang et al. (2023) traz uma breve definição para o bloco: uma sequência de convoluções com filtros 3×3 com *padding* 1, altura e largura são mantidas, seguida por uma camada de *max-pooling* 2×2 com *stride* de 2, dividindo ao meio

altura e largura após cada bloco. Os filtros são ditos de dimensão baixa, pois tem a menor medida capaz de guardar as noções de esquerda/direita, topo/base e centro. Dentre o conjunto de configurações, há as mais bem-sucedidas: D e E, popularmente conhecidas como VGG-16 e VGG-19 (KHAN et al., 2018).

A arquitetura pode ser separada em duas seções principais: a primeira constituída principalmente dos blocos mencionados anteriormente, e a segunda formada por camadas *fully connected* (ZHANG et al., 2023). O modelo VGG original tinha cinco blocos convolucionais, onde os dois primeiros tinham uma camada convolucional cada e os três últimos duas camadas convolucionais, seguidos de três camadas *fully connected*. Em outras palavras, o modelo original era organizado em onze camadas onde os pesos eram ajustáveis, e por esse motivo, chamado VGG-11. Na Figura 2.10 podemos ver em detalhes a organização de VGG-16. Há cinco blocos com camadas de convolução, os dois primeiros, cada, usam duas camadas de convolução e os três últimos três, seguidos de três camadas *fully connected*, somando 16 camadas onde os pesos são ajustáveis.

Para o treinamento, cada imagem tem um tamanho fixo de 224×224 pixels e há um pré-processamento onde o valor RGB médio, computado durante o treinamento, é subtraído de cada pixel. E então cada imagem passa pela pilha de blocos sucessivamente. Khan et al. (2018) ressalta o uso de filtros de baixa dimensão nos blocos como uma boa escolha no projeto da arquitetura, pois isso conduz a um número reduzido de parâmetros, 138 milhões para a VGG-16, sucedendo em treinamento e teste eficientes, além de melhor performance para tarefas de visão computacional.

O trabalho de Simonyan e Zisserman (2014) mostra o potencial de generalização do modelo, exibindo resultado satisfatórios quando submetido a alguns desafios, tais como o VOC-2012 (EVERINGHAM et al., 2015) e o Caltech-256 (GRIFFIN; HOLUB; PERONA, 2007).

2.4.2 ResNet-152

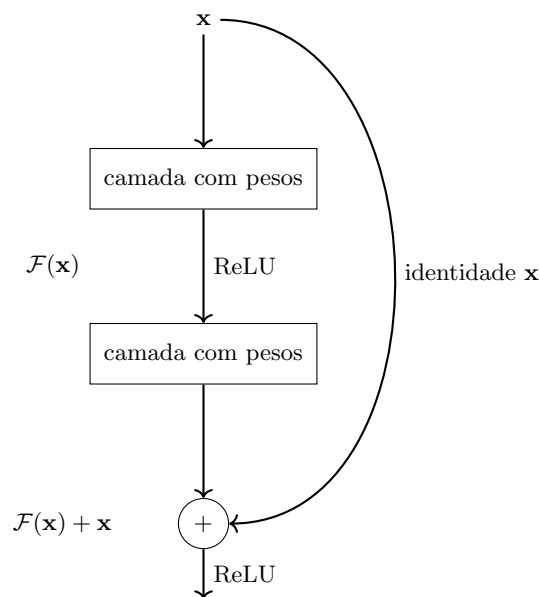
Depois do sucesso da VGG-16, o trabalho de He et al. (2016) começa com a pergunta: *Aprender redes melhores é tão simples quanto empilhar mais camadas?*. Apesar dos obstáculos para

evidente: à medida que a profundidade da rede aumenta, a acurácia se satura e então se degrada rapidamente.

Durante seus experimentos He et al. (2016) observaram que a degradação da acurácia não era causada por *overfitting*, e adicionar mais camadas a um modelo já suficientemente profundo levava a um erro de treinamento maior. Para superar tal dificuldade um novo *framework* foi indicado: o aprendizado residual profundo. O novo paradigma torna compreensível o prefixo ResNet, do inglês *residual networks*, ou redes residuais.

A Figura 2.11 exibe uma representação gráfica da proposta do bloco residual de He et al. (2016). Seja $\mathbf{x}$ a entrada. O bloco força o aprendizado do resíduo $\mathcal{F}(\mathbf{x}) - \mathbf{x}$, dando origem ao termo residual. Se o mapeamento de identidade $\mathcal{F}(\mathbf{x}) = \mathbf{x}$ é a representação subjacente aprendida, a diferença (ou resíduo), é mais fácil de aprender: basta levar os pesos e vieses das camadas dentro do bloco a zero (ZHANG et al., 2023).

Figura 2.11: Bloco residual usado no aprendizado residual profundo.



Fonte: Adaptado de HE et al. (2016, p.2)

Para Khan et al. (2018), as conexões de salto observadas no bloco residual indicam a característica mais importante das redes residuais, pois facilita o treino de arquiteturas profundas de CNNs. Usando o bloco residual, He et al. (2016) ganharam o ILSVRC no ano de 2015 com

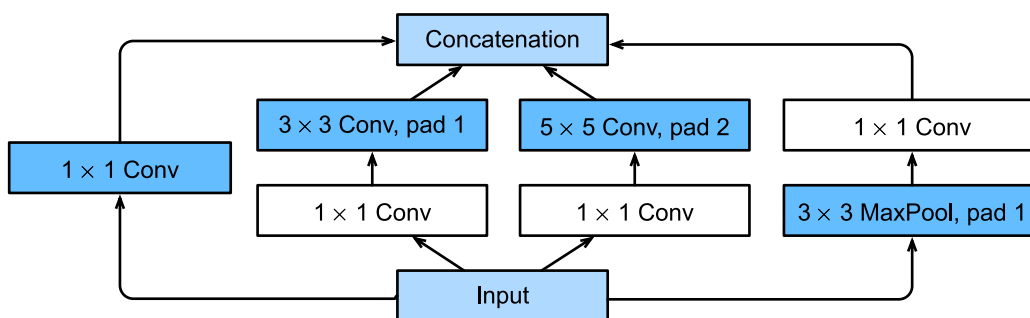
um modelo composto de 152 camadas de pesos ajustáveis, a ResNet-152. Além da vitória no ILSVRC em 2015, o modelo também demonstrou generalização em outras tarefas de reconhecimento, e ganhou o primeiro lugar em outros desafios: detecção no ImageNet, localização no ImageNet, detecção no COCO (LIN et al., 2014) e segmentação no COCO no mesmo ano.

2.4.3 Inception-v3

Um ano antes da vitória de ResNet, GoogLeNet foi o modelo vencedor do ILSVRC em 2014 (KHAN et al., 2018). Nas palavras de Zhang et al. (2023), o modelo poderia ser descrito como uma combinação das vantagens da NiN (*Network in Network*) (LIN; CHEN; YAN, 2013), blocos repetidos (SIMONYAN; ZISSERMAN, 2014) (VGG) e um coquetel de filtros de convolução. Ainda segundo ZHANG et al. a notável contribuição de Szegedy et al. (2015) foi o projeto do corpo da rede, pois resolvia o problema de seleção dos filtros convolucionais de um jeito simples.

A estrutura do bloco principal de GoogLeNet é exibida na Figura 2.12. Há quatro ramos paralelos que fazem uso de filtros convolucionais de tamanhos distintos a fim de extrair informações de diferentes dimensões espaciais. Zhao et al. (2024) resume bem o propósito do bloco: extrair informações de características das camadas, fundi-las e passá-las para as camadas subsequentes.

Figura 2.12: Estrutura do bloco Inception



Fonte: ZHANG et al. (2023, p. 272).

Um ponto enfatizado pelos próprios idealizadores de Inception é a quantidade de parâmetros na rede comparado a outras arquiteturas. Apesar da rede ser mais profunda que a VGG, por exemplo, possui menos parâmetros que ela. De fato, seu custo computacional é menor do que

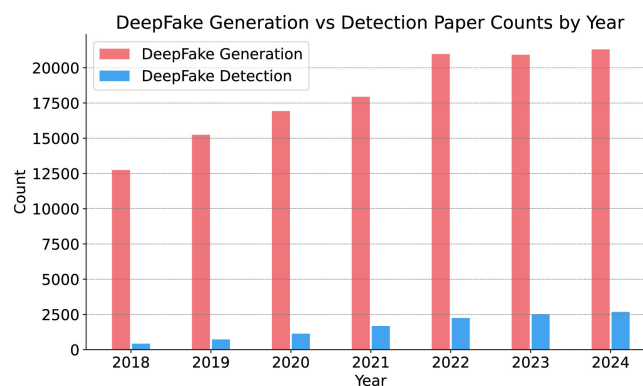
o de seus antecessores, ao mesmo tempo que oferece maior precisão (ZHANG et al., 2023). Ou seja, na conclusão de Zhao et al. (2024), quando os recursos de hardware de um computador são limitados, Inception é uma solução superior para classificação de imagens.

Depois do lançamento de Inception-v1, Szegedy et al. propuseram melhorias na estrutura do bloco Inception e a Inception-v2 foi lançada (SZEGEDY et al., 2016). A Inception-v3 é resultado de otimizações das Inception v1 e v2. A ideia principal era reduzir o custo computacional da rede profunda sem comprometer a capacidade de generalização (ZHAO et al., 2024).

2.5 Trabalhos Relacionados

De acordo com Xie et al. (2025), o principal foco de pesquisa sobre *deepFakes* está nos rostos. Os autores apontam que a provável razão para tal é o fato de que os rostos são informações de identificação pessoal, e o dano associado à síntese ou manipulação deste tipo de conteúdo é grave. Como resultado, a pesquisa em geração de *deepfakes* está sempre à frente de pesquisas em detecção. Esse dado é bem ilustrado na Figura 2.13.

Figura 2.13: Estatísticas de artigos publicados sobre geração e detecção de DeepFake. Dados obtidos da Web of Science, até dezembro de 2024



Fonte: XIE et al. (2025, p. 2).

Para contextualizar o cenário experimental desta pesquisa, realizou-se uma busca por trabalhos da literatura que citam a obra de Cheng et al. (2024). Os critérios de seleção delimitaram o escopo a investigações que utilizam o conjunto de dados DiFF para a detecção de *deepfakes*

faciais em imagens. A seguir serão apresentados três contribuições recentes, caracterizando principalmente o uso de DiFF e então como o trabalho apresentado aqui se relaciona as contribuições listadas.

A proposta de Wang et al. (2025) é um *self-supervised pretraining framework* para aprender as características essenciais de faces reais. Para validar essa proposta, Wang et al. (2025) faz uso de DiFF a fim de investigar a capacidade de adaptação e generalização do modelo frente aos métodos de geração de *deepfakes* que estão em constante evolução. O modelo é pré-treinado com a versão comprimida c23 de FaceForensics++ (FF++) (ROSSLER et al., 2019) e testado contra os subconjuntos do DiFF. Os autores ressaltam o desafio nesse experimento em relação aos outros experimentos realizados no trabalho usando como justificativa a diferença entre os dados do conjunto de teste e os métodos usados para geração das imagens sintetizadas. Os resultados são reportados usando a métrica AUC para cada subconjunto. A média relatada, considerando os subconjuntos, foi 70.99.

Ma et al. (2026) também usaram DiFF com o intuito de validar a sua proposta de contribuição. Nesse caso a ideia principal era identificar as principais e mais comuns características dentre os vários tipos de *deepfake*. Da mesma maneira, o modelo foi treinado na versão comprimida de FF++ e testado usando DiFF mediante seus subconjuntos. Um ponto distinto a se assinalar é a métrica de resultado AUC a nível de frame. A média dos subconjuntos foi 83.28.

Já no trabalho de Chen et al. (2026) o objetivo foi mais abrangente na redação em relação aos mencionados anteriormente: um *framework explainable e extendable* baseado em *multimodal large-language models* para detecção de *deepfake* (χ^2 -DFD). Assim como seus antecessores, o modelo foi treinado com c23 de FF++ e testado atendendo a DiFF. A métrica principal de avaliação também é AUC, mas dessa vez nem todos os subconjuntos são usados. A melhor média de AUC, considerando três de quatro subconjuntos, foi 90.96.

A Tabela 2.1 sintetiza os três trabalhos apontados, destacando o conjunto de treinamento FF+ e o uso do conjunto de dados DiFF para testes. Ressalta-se o desempenho com a métrica AUC e a ausência de transferência de aprendizado. No escopo do presente trabalho, será explorada a estratégia da Transferência de Aprendizado que, conforme reportado na Seção

Tabela 2.1: Análise comparativa de métodos recentes de detecção de *deepfakes*, destacando os conjuntos de dados de treinamento e teste, a métrica de Melhor AUC (%) e a utilização de aprendizado de transferência.

Trabalho	Treinamento	Teste	AUC (%)	Transferência de Aprendizado?
Wang et al. (2025)	FF++	DiFF	70,99	Não
Ma et al. (2026)	FF++	DiFF	83,28	Não
Chen et al. (2026)	FF++	DiFF*	90,96	Não

* O framework utiliza um subconjunto específico a partir do conjunto de dados original.

2.3, mostra-se útil para reaproveitamento de aprendizado de características entre domínios diferentes, diminuindo o ônus de treinamento. Além disso, será avaliado apenas a base de dados DiFF como fonte de experiência, por essa ser mais recente na literatura, contemplando as estratégias mais recentes de geração de *deepfakes*.

Capítulo 3

Material e Métodos

Este estudo investigou a aplicação de transferência de aprendizado em CNNs na tarefa de Visão Computacional de detecção de *deepfakes* faciais, problema este abordada como uma tarefa de classificação binária mediante o paradigma de Aprendizado Supervisionado. Para construir a solução proposta, a Seção 3.1 detalha os dados experimentais empregados na coleta de conhecimento; a Seção 3.2 apresenta como os dados foram preparados; a Seção 3.3 descreve os modelos e os experimentos propostos; a Seção 3.4 descreve a obtenção das métricas de desempenho; e a Seção 3.5 descreve os recursos computacionais utilizados no trabalho.

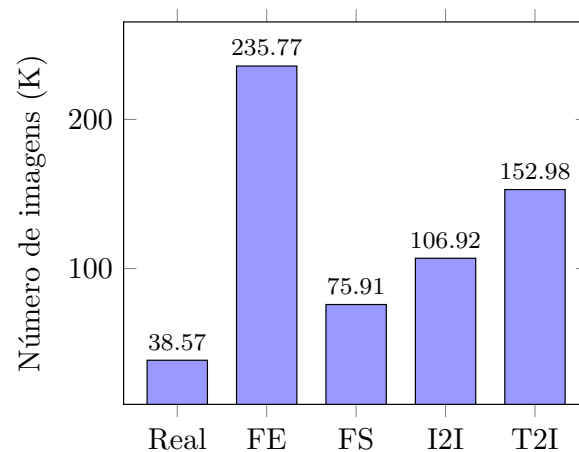
3.1 Dados Experimentais

O conjunto de dados usado para fins experimentais neste trabalho foi introduzido por Cheng et al. (2024) e é composto de imagens em duas categorias principais: reais, identificando 1070 celebridades, e sintetizadas (*deepfakes*), geradas a partir das imagens reais usando modelos de difusão. Somadas as duas classes de imagens, o número total de exemplos era 610.143.

Os modelos de difusão podem ser classificados em quatro categorias segundo o tipo de entrada: Text-to-Image (T2I), que leva em conta prompts de texto; Image-to-Image (I2I) e Face Swapping (FS), onde as entradas são visuais e, Face Editing (FE) que usa uma combinação de entrada visual e textual. A Figura 3.1 apresenta o número total de imagens reais e sintetizadas, as últimas classificadas de acordo com o tipo de entrada. É possível verificar que o

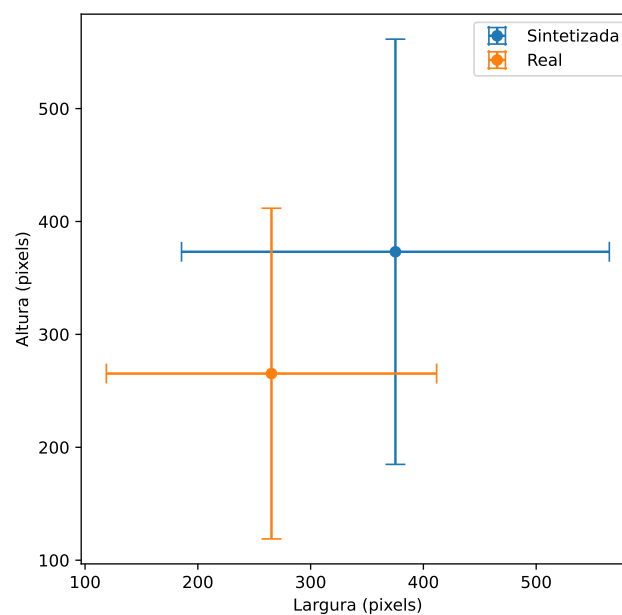
conjunto de dados era composto principalmente por imagens sintetizadas, o que caracteriza o seu desbalanceamento.

Figura 3.1: Distribuição de imagens do conjunto de dados por tipo



A representação da média de altura e largura, em pixels, dos subconjuntos de imagens reais e sintetizadas pode ser vista na Figura 3.2. Ambos subconjuntos não tem variação significativa nas dimensões, como indicam as barras. As imagens reais são menores em relação as sintetizadas.

Figura 3.2: Distribuição média das dimensões das imagens. O ponto representa a média da largura e da altura, enquanto as barras indicam o desvio padrão em cada dimensão.



O conteúdo das imagens se caracterizava pela face de uma pessoa sobre um fundo qualquer. A Figura 3.3 ilustra cinco imagens randomicamente selecionadas dos subconjuntos de imagens reais e sintetizadas. É importante destacar que alguns dos *deepfakes* da base de dados possuem alta verossimilhança com imagens de pessoas reais, o que denota a dificuldade prática da tarefa em questão.

Figura 3.3: Imagens reais (primeira linha) e imagens sintetizadas (segunda linha).



Como descrito anteriormente, inicialmente, os dados estavam divididos em subconjuntos. Cada subconjunto estava dividido em outros três: treinamento, teste e validação. Na Figura 3.4 é observado que a proporção dessa divisão, em relação as classes reais e sintetizadas, era diferente. Enquanto o subconjunto de imagens reais tinha um número maior concentrado em teste, essa afirmação era verdade para a partição de treinamento quando falamos do subconjunto de imagens sintetizadas. Já dentro do subconjunto de imagens sintetizadas, mediante as partições de treinamento, teste e validação e classificação dos modelos de difusão, a distribuição é proporcional (Figura 3.5), e como descrito anteriormente na Figura 3.4, essa distribuição era desigual para o subconjunto de imagens reais. Partindo dessa condição uma nova estratégia de partição para este trabalho foi tomada e é detalhada na próxima seção.

3.2 Preparação das Amostras

Para esse estudo escolhemos usar uma divisão para treinamento, teste e validação na proporção 7:2:1, segundo o método de validação cruzada *holdout*. Para lidar com as diferenças no quantitativo dos tipos de imagem, as imagens reais e sintetizadas foram tomadas em igual proporção

Figura 3.4: Distribuição de imagens do conjunto de dados por partição

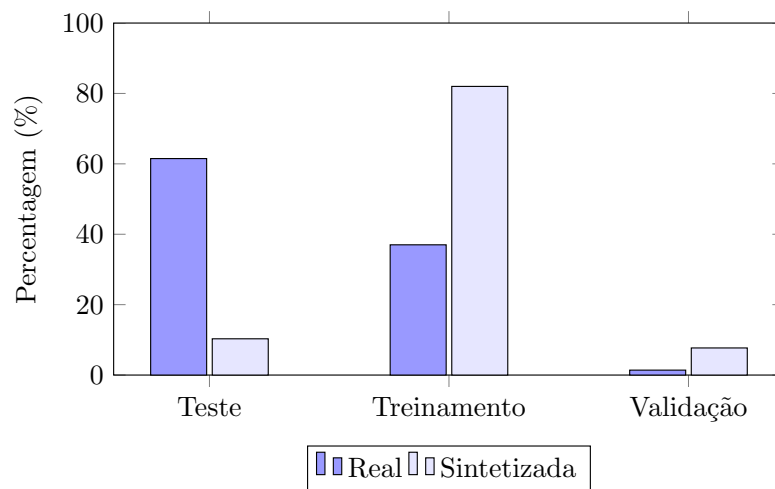
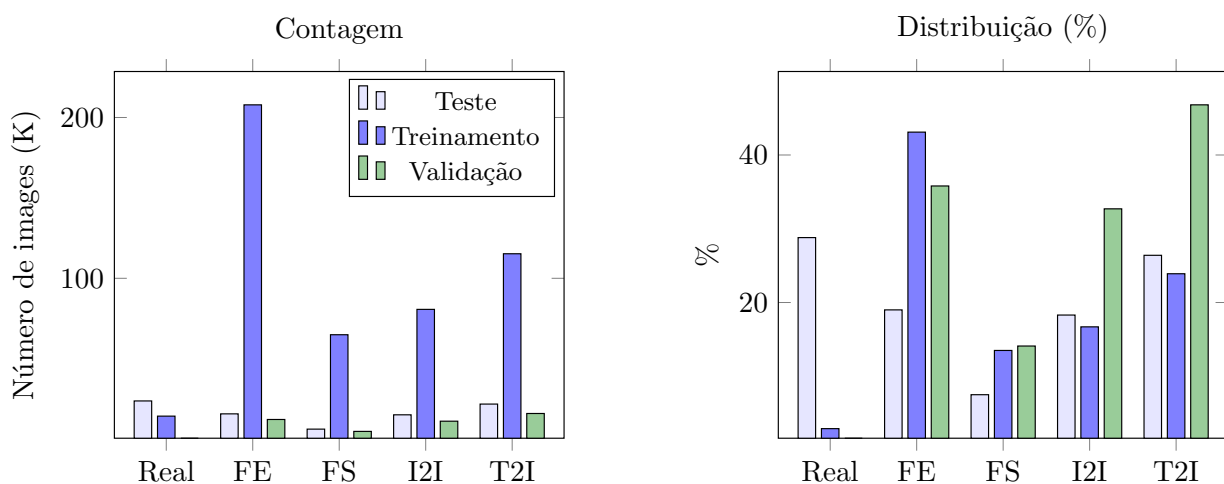


Figura 3.5: Distribuição de imagens do conjunto de dados por partição e tipo



1:1, ou seja, optou-se pela subamostragem das classes sintéticas para fins de balanceamento. À medida que as imagens sintetizadas também podiam ser distintas em relação ao método de geração, essas foram selecionadas em igual medida, conforme Figura 3.6. Ao final, tem-se uma base de dados balanceada para o problema em questão, cuja distribuição ao longo das partições encontra-se na Figura 3.7.

Figura 3.6: Composição dos conjuntos de amostras de treinamento, teste e validação

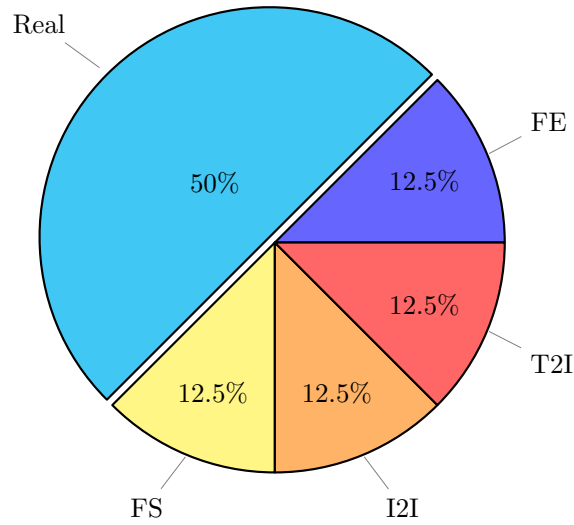
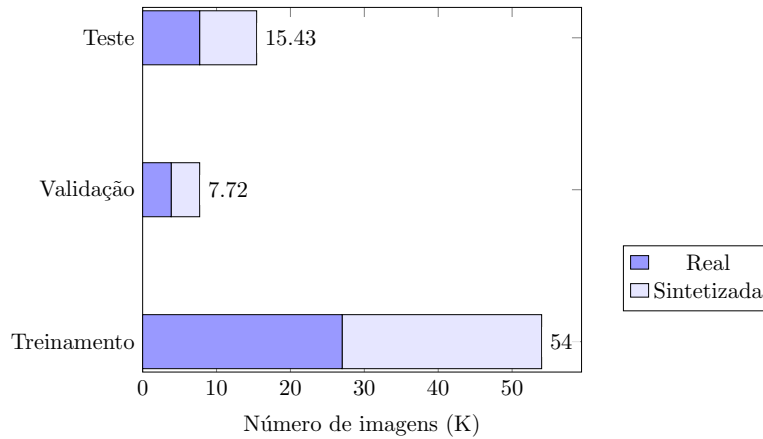


Figura 3.7: Contagem do número de imagens segundo partição e classe



3.3 Modelos, Parâmetros e Hiperparâmetros

Três modelos de redes neurais convolucionais foram usados nos experimentos realizados no escopo deste trabalho, a citar: VGG-16, ResNet-152 e Inception-v3. As implementações dos modelos foram obtidas junto à biblioteca `torchvision`, parte do projeto PyTorch. Os experimentos foram divididos em dois conjuntos que serão descritos a seguir.

3.3.1 Experimento 1

No Experimento 1 utilizou-se a técnica de *feature extraction*. Todos os modelos tiveram todas as suas camadas congeladas, exceto a camada final. Nesse caso os pesos das camadas congeladas

não são atualizados. Para cada modelo os pesos foram inicializados com opções disponibilizadas pela própria biblioteca após treinamento com ImageNet. A camada final também foi substituída, dado que a tarefa alvo desta pesquisa é diferente da tarefa de origem para a qual os modelos foram propostos. A Tabela 3.1 mostra em detalhes parâmetros e hiperparâmetros das três configurações do Experimento 1.

Tabela 3.1: Especificações de parâmetros e hiperparâmetros do Experimento 1 por modelo

Modelo	Parâmetros	Otimizador	Taxa de aprendizado
VGG-16	8.19 k	NAdam	1×10^{-4}
ResNet-152	4.10 k	NAdam	1×10^{-4}
Inception-v3	5.64 k	NAdam	1×10^{-4}

3.3.2 Experimento 2

No Experimento 2 utilizou-se *fine-tuning*. Além da camada final, que também foi substituída como no Experimento 1, as camadas nos últimos blocos não foram congeladas. Os pesos foram inicializados como no Experimento 1. Parâmetros e hiperparâmetros das configurações relativas ao Experimento 2 são descritas na Tabela 3.2.

Tabela 3.2: Especificações de parâmetros e hiperparâmetros do Experimento 2 por modelo

Modelo	Parâmetros	Otimizador	Taxa de aprendizado
VGG-16	7.09 M	SGD	1×10^{-3}
ResNet-152	14.97 M	SGD	1×10^{-3}
Inception-v3	12.83 M	SGD	1×10^{-4}

3.4 Avaliação de Desempenho

Para avaliar os modelos selecionados perante o problema proposto, considerou-se uma validação cruzada do tipo *holdout*, com 70 % dos dados disponíveis para treino, 10 % para validação e 20 % para testes, em que esta última partição foi utilizada para aferir o desempenho. As métricas a seguir foram elencadas para quantificar o desempenho dos modelos:

$$\text{Acurácia} = \frac{VP + VN}{VP + VN + FP + FN} \quad (3.1)$$

$$\text{Precisão} = \frac{VP}{VP + FP} \quad (3.2)$$

$$\text{Revocação} = \frac{VP}{VP + FN} \quad (3.3)$$

$$F_1 - \text{Score} = 2 \left(\frac{\text{Precisão} \cdot \text{Revocação}}{\text{Precisão} + \text{Revocação}} \right) \quad (3.4)$$

Nas Eqs. (3.1)-(3.4), as siglas indicam os quatro resultados possíveis de uma tarefa de classificação binária, a citar: VP denota o quantitativo das classificações corretas para a classe positiva; VN, as classificações corretas para a classe negativa; FP, os Erros Tipo I; e FN, os Erros Tipo II.

3.5 Ambiente de Desenvolvimento

O ambiente de desenvolvimento consistiu em um servidor computacional equipado com processador Intel® Core™ i9-10900K com clock de 3,7 GHz, memória principal de 64 GB, 10 TB de memória secundária e 3 placas de vídeo NVIDIA GeForce RTX 3060 com 12 GB de VRAM cada para aceleração em *hardware* do treinamento, operando com o *driver* versão 560.35.03 e suporte à API CUDA versão 12.6.

Capítulo 4

Resultados e Discussão

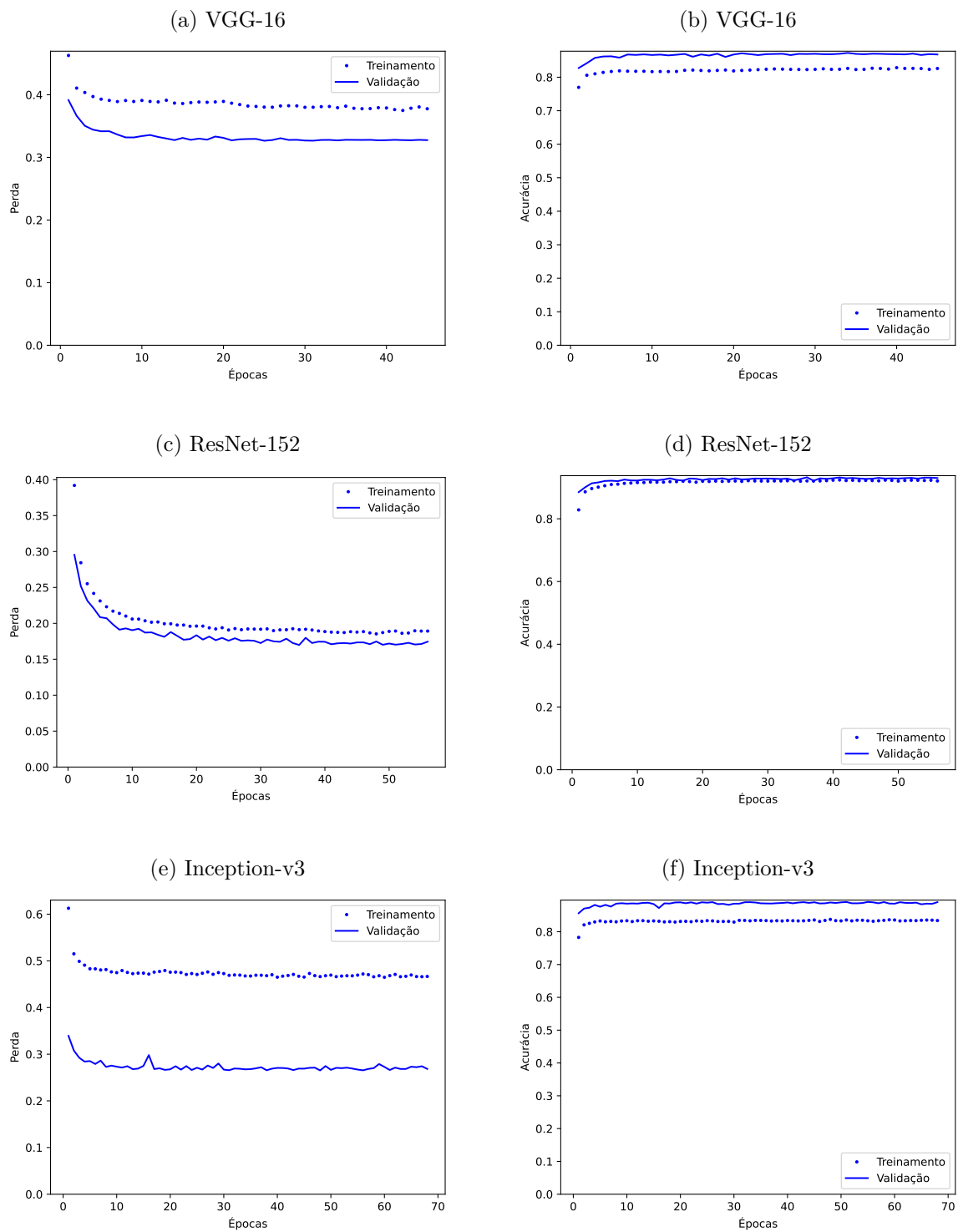
Esse capítulo contempla a apresentação dos resultados dos experimentos realizados, juntamente com uma análise crítica e comparativa dos resultados. Além dos experimentos previstos, efetuou-se uma validação externa dos melhores modelos afim de aferir a capacidade de generalização perante outro *dataset* de *deepfakes* faciais existentes na literatura, contribuindo para a discussão dos desafios da detecção desse tipo de mídia sintética.

4.1 Resultados do Experimento 1

Ao longo das épocas de treinamento dos modelos, foram coletadas métricas de perda e acurácia perante o conjunto de treino e de validação, conforme ilustrado na Figura 4.1. Nota-se que as curvas de perda em treinamento e validação são mais próximas para VGG-16 (4.1a) e ResNet-152 (4.1c) em relação a Inception-v3 (4.1e). O aspecto das curvas de acurácia é semelhante e o que as distingue é a quantidade de épocas que o treinamento levou para ser concluído. Dito isso, VGG-16 foi a mais rápida a convergir e a Inception-v3 foi a CNN que teve um treinamento mais longo no que diz respeito ao total de épocas.

Conforme exposto na Tabela 4.1, onde são informados precisão, revocação, F1-Score e acurácia para o conjunto de testes, verifica-se que ResNet-152 obteve o melhor desempenho no Experimento 1. Isso pode ser corroborado pelos números das matrizes de de confusão na Figura 4.2. As matrizes mostram que os modelos tenderam a errar mais os rostos reais. Olhando

Figura 4.1: Curvas de perda (à esquerda) e acurácia (à direita) de treinamento e validação para o Experimento 1 (*feature extraction*).



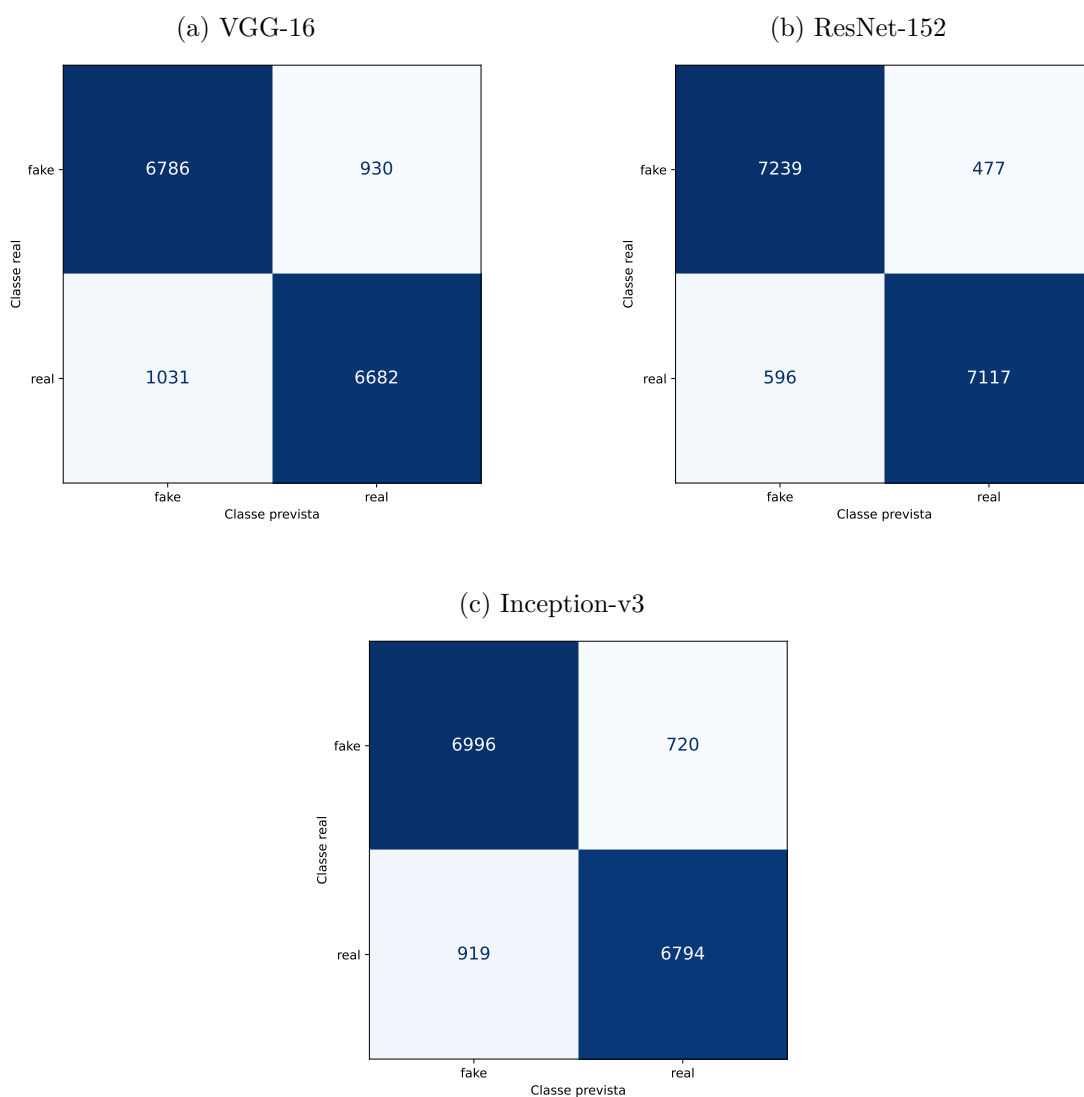
os falsos negativos, onde a classe prevista foi real mas na verdade se tratava de um exemplo de *deepfake*, VGG-16 (4.2a) foi a que teve o pior desempenho. ResNet-152 (4.2b) teve mais êxito

ao classificar ambas as classes sem a necessidade de alteração de seus pesos internos.

Tabela 4.1: Métricas de desempenho obtidas no conjunto de teste.

Modelo	Precisão	Revocação	F1-Score	Acurácia
VGG-16	0.8729	0.8729	0.8729	0.8729
ResNet-152	0.9305	0.9304	0.9304	0.9304
Inception-v3	0.8940	0.8937	0.8937	0.8937

Figura 4.2: Matrizes de confusão obtidas no conjunto de teste para o Experimento 1 (*feature extraction*).



Com o intuito de evidenciar qualitativamente o desempenho de ResNet-152, foram selecionadas amostras das imagens pertencentes à classe sintetizada (*deepfake*), as quais tiveram

predição correta de ResNet-152, mas incorreta pela VGG-16 e Inception-v3. Essa seleção contemplou uma imagem de cada um dos quatro subconjuntos que compõem a referida classe, ilustradas na Figura 4.3. Destaca-se a capacidade da ResNet-152 no Experimento 1 em classificar corretamente *deepfakes* originários de diferentes métodos de síntese.

Figura 4.3: Exemplos visuais da classe sintetizada classificados corretamente de forma exclusiva por ResNet-152 no Experimento 1.



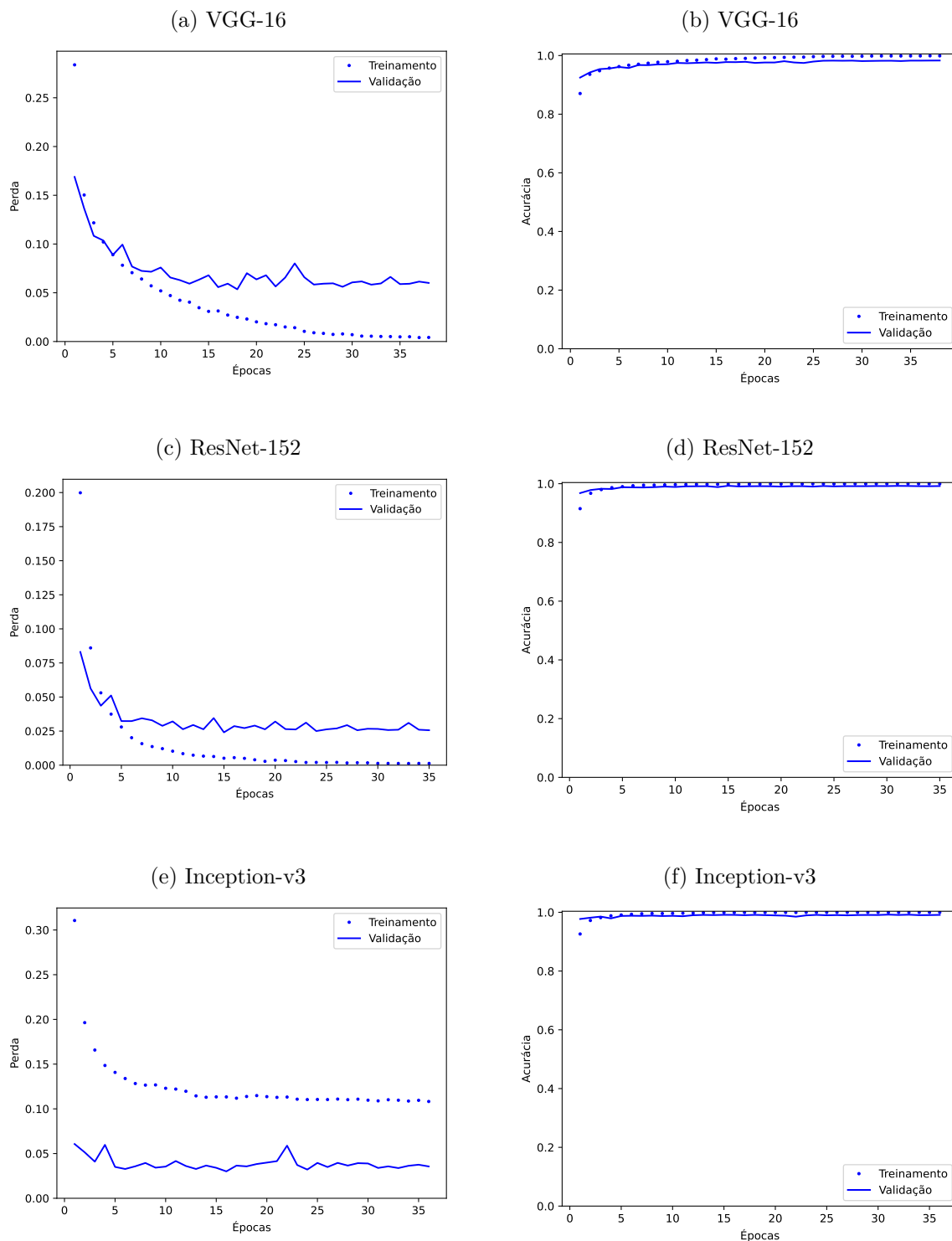
4.2 Resultados do Experimento 2

Assim como no Experimento 2, o treinamento das redes foi monitorado ao longo das épocas, conforme Figura 4.4. Aqui, todas as curvas de perda demonstram variação maior em vista do maior número de pesos treináveis. De modo semelhante ao Experimento 1, as curvas de perda das fases de treinamento e validação de Inception-v3 (4.4e) tem um distanciamento desigual quando comparado a VGG-16 (4.1a) e ResNet-152 (4.4c). Em contrapartida, as curvas de acurácia são análogas em formato nas três configurações. Também é notado que o número total de épocas é próximo considerando os três modelos, diferentemente do Experimento 1. No contexto desse experimento se observou que a estratégia de *early stopping* foi determinante para evitar a transferência negativa, haja vista a distância entre as curvas de perda de treinamento e de validação dos três modelos no final do treinamento.

As mesmas métricas de desempenho para o Experimento 1 foram coletadas no Experimento 2, vide Tabela 4.2. Aqui, considerando o descongelamento das camadas finais de cada modelo, o desempenho das arquiteturas se aproxima e a maior diferença entre os números de acurácia passou a ser de pouco mais de 0.5%.

Enquanto no Experimento 1 todos os modelos tiveram mais dificuldade para identificar

Figura 4.4: Curvas de perda (à esquerda) e acurácia (à direita) de treinamento e validação para o Experimento 2 (*fine-tuning*).

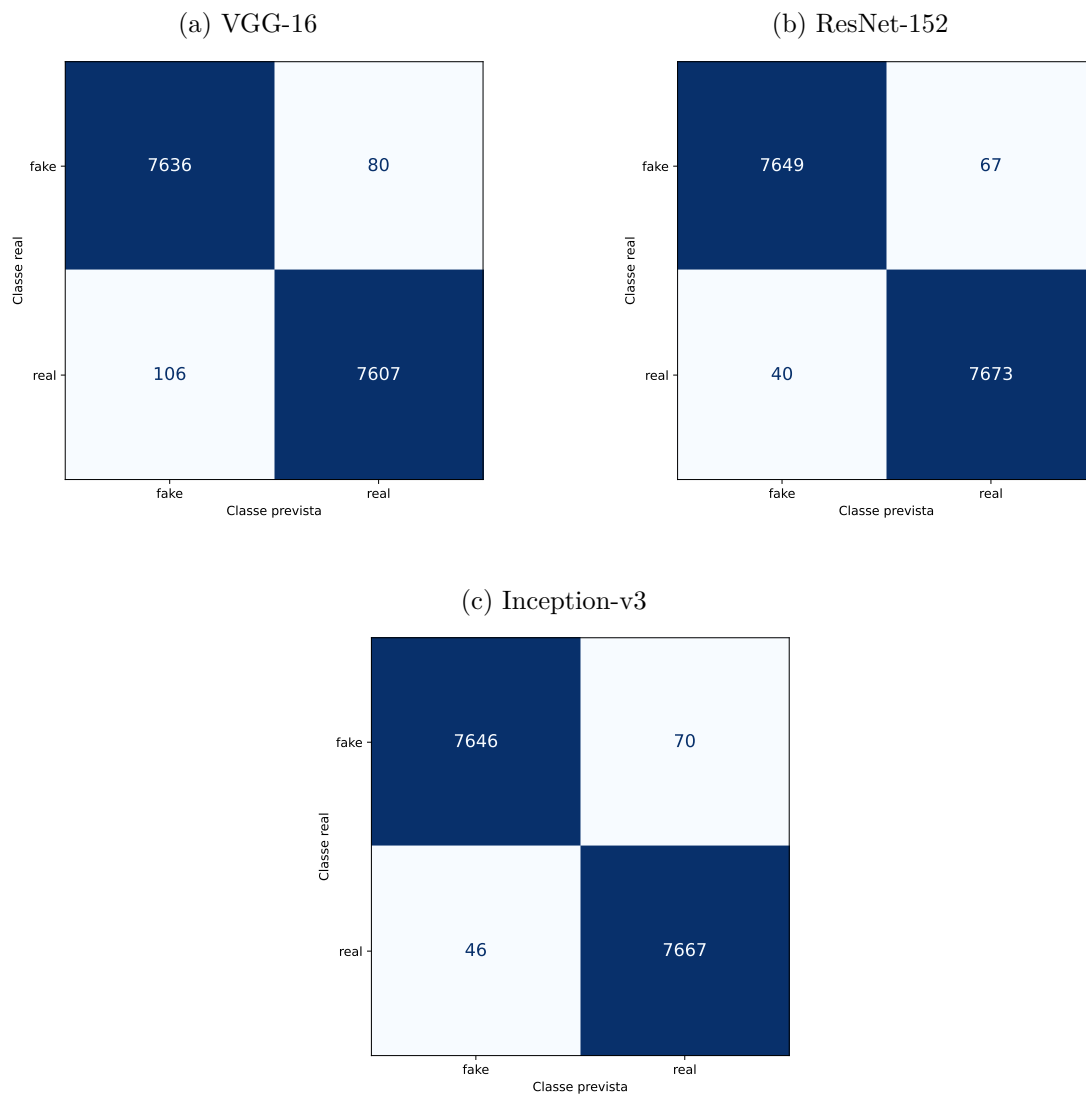


rostos reais, no Experimento 2 apenas VGG-16 demonstrou comportamento similar. Mesmo assim, o número de falsos negativos é baixo e possibilita afirmar que o descongelamento permitiu

Tabela 4.2: Métricas de desempenho obtidas no conjunto de teste no Experimento 2.

Modelo	Precisão	Revocação	F1-Score	Acurácia
VGG-16	0.9879	0.9879	0.9879	0.9879
ResNet-152	0.9930	0.9930	0.9930	0.9930
Inception-v3	0.9924	0.9924	0.9924	0.9924

aos modelos detectar melhor as características dos exemplos do tipo *deepfake*, conforme matrizes de confusão demonstradas na Figura 4.5.

Figura 4.5: Matrizes de confusão obtidas no conjunto de teste para o Experimento 2 (*fine-tuning*).

Apesar de todas as arquiteturas analisadas apresentarem um desempenho com acurácias

superiores a 99%, buscou-se compreender os limites de generalização dos modelos. Para tanto, isolaram-se as imagens que nenhum dos modelos foi capaz de classificar corretamente. A Figura 4.6 apresenta uma colagem dessas imagens oriundas da partição de testes. Percebe-se que a verossimilhança de algumas dessas imagens torna a tarefa realmente desafiadora.

Figura 4.6: Colagem de imagens classificadas incorretamente como reais por todas as arquiteturas avaliadas no Experimento 2



4.3 Análise Comparativa dos Experimentos

A Tabela 4.3 exibe as métricas dos testes realizados em ambos os experimentos. A melhor marca de revocação foi de ResNet-152 no Experimento 2. O modelo em questão também obteve a maior acurácia e a melhor F1-Score nos dois experimentos. Deve-se notar os valores iguais na Tabela 4.3 exceto a precisão de Inception-v3 no Experimento 1. Isso deve ser atribuído ao balanceamento do conjunto de teste.

O tempo de treinamento e o número de parâmetros são dispostos na Tabela 4.4. Convertido para horas, o tempo de treinamento total pode ser comparado na Figura 4.7a. No Experimento 1, o modelo mais pesado, em termos de parâmetros, foi o VGG-16. No entanto quem se destacou com o maior tempo de treinamento foi ResNet-152. A performance do modelo, sendo melhor no Experimento 1, justificou o ônus de tempo de treinamento em relação aos outros dois.

Embora o descongelamento das camadas no Experimento 2 pudesse indicar que o tempo de

Tabela 4.3: Métricas de desempenho obtidas no conjunto de teste nos Experimento 1 e 2

Modelo	Experimento	Precisão	Revocação	F1-Score	Acurácia
VGG-16	1	0.8729	0.8729	0.8729	0.8729
	2	0.9879	0.9879	0.9879	0.9879
ResNet-152	1	0.9305	0.9304	0.9304	0.9304
	2	0.9930	0.9930	0.9930	0.9930
Inception-v3	1	0.8940	0.8937	0.8937	0.8937
	2	0.9924	0.9924	0.9924	0.9924

treinamento seria mais longo, isso não ocorreu, conforme Figura 4.7a. A média de tempo por época é superior devido o número superior de pesos treináveis, tal como ilustrado na Figura 4.7b. Mesmo assim a convergência aconteceu antes e encurtou a duração do estágio. A diferença foi maior para ResNet-152 e Inception-v3.

Apesar do número de parâmetros, Inception-v3 foi o modelo que apresentou o melhor custo-benefício, tempo de treinamento e acurácia (quando apresentado ao conjunto de teste).

Tabela 4.4: Tempo de treinamento e total de parâmetros por experimento.

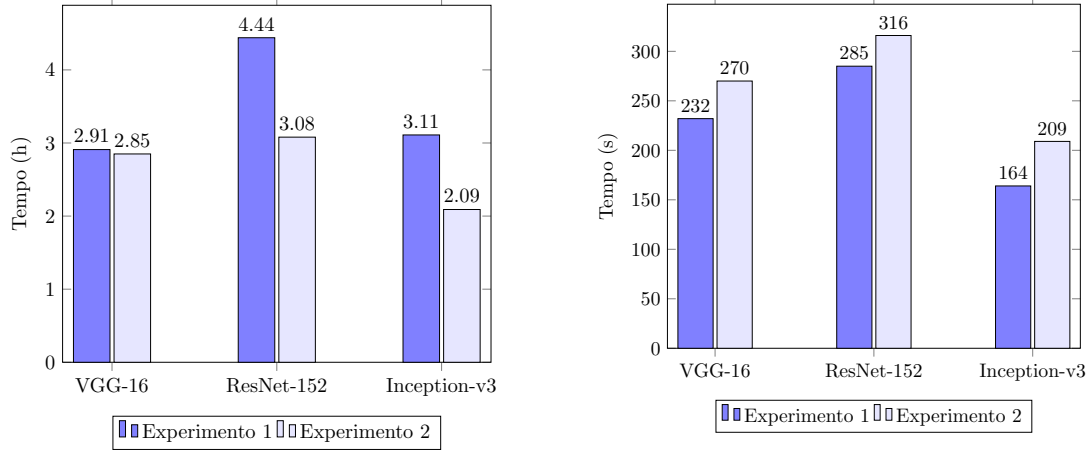
Modelo	Experimento	Tempo de treinamento (s)	Parâmetros
VGG-16	1	10478	8.19 k
	2	10260	7.09 M
ResNet-152	1	15998	4.10 k
	2	11079	14.97 M
Inception-v3	1	11184	5.64 k
	2	7535	12.83 M

Com a finalidade de compreender a robustez dos modelos, os erros na classe *fake* foram segmentados de acordo com a técnica usada para gerar tal exemplo. Isso é exibido na Tabela 4.5. Como antes mencionado, ResNet-152 foi o modelo que demonstrou melhor desempenho nos dois experimentos. Esse fato também pode ser notado na tabela, pois ResNet-152 foi o modelo que teve menor redução no número de erros. Por esse motivo, mostramos a distribuição de erros nos Experimentos 1 e 2 por subconjunto da classe sintetizada (*fake*) na Figura 4.8.

Na Figura 4.8a é possível observar maior dificuldade do modelo ao identificar exemplos do subconjunto onde as imagens são oriundas de *Face Swapping* (FS), mas isso não se repete no

Figura 4.7: Comparativo do custo temporal dos experimentos

(a) Tempo total de treinamento acumulado em horas (b) Tempo médio de execução por época em segundos

Tabela 4.5: Distribuição absoluta de falsos negativos (erros) cometidos pelos modelos em função de subconjunto da classe sintetizada (*fake*)

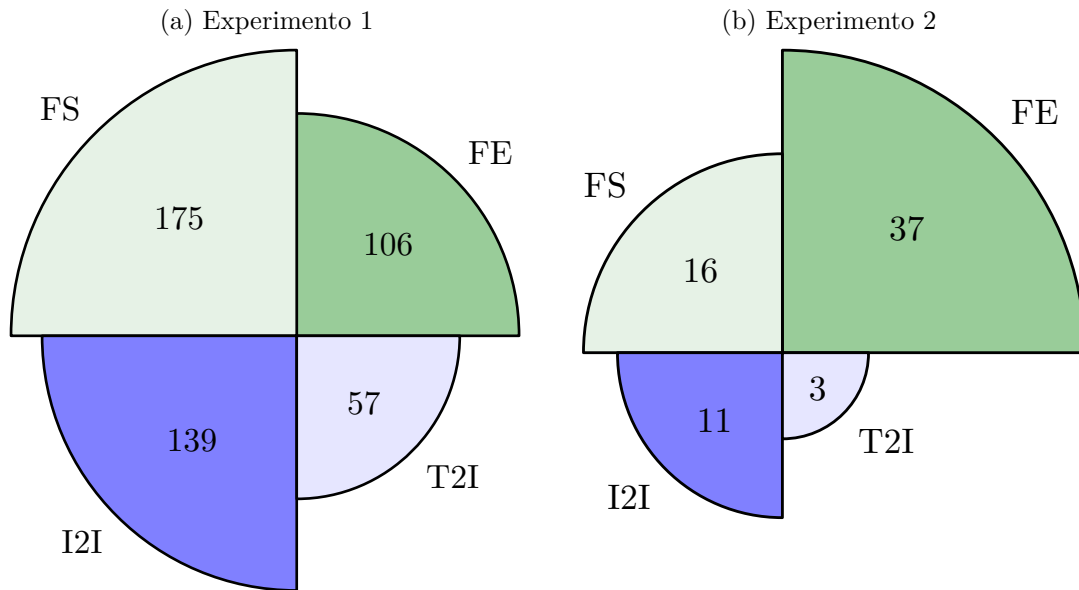
Modelo	Experimento	FE	FS	I2I	T2I	Total de Erros	Redução (%)
VGG-16	1	181	407	179	163	930	—
	2	49	8	16	7	80	91,40%
ResNet-152	1	106	175	139	57	477	—
	2	37	16	11	3	67	85,95%
Inception-v3	1	185	256	186	93	720	—
	2	49	12	5	4	70	90,28%

Experimento 2, conforme Figura 4.8b. Ao descongelar algumas camadas, o subconjunto *Face Editing* (FE) se mostrou mais resistente a classificação. Destaca-se também a proporção similar dentre os erros dos outros subconjuntos.

4.4 Validação Externa

A fim de testar a capacidade de generalização dos melhores modelos obtidos neste experimento, foram executados testes usando imagens de outro domínio, com vistas a realizar uma validação externa. As imagens selecionadas pertenciam ao DF-40 (YAN et al., 2024), um conjunto de dados pensando especialmente para detecção de *deepfakes*. A proposta de Yan et al. (2024) tem exemplos reais e sintetizados. Esses últimos são gerados a partir de 40 técnicas distintas, o

Figura 4.8: Distribuição de erros por subconjuntos da classe sintetizada para o modelo ResNet-152 por experimento.



que origina o nome do *dataset*, compondo um conjunto de dados do estado da arte em geração de faces sintéticas.

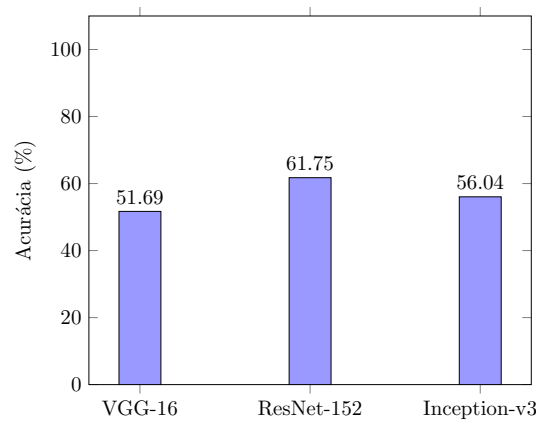
A partir dos melhores modelos alcançados no Experimento 2, foram executados novamente previsões em dois conjuntos de testes. Nestes, os dados reais são iguais. As amostras da classe de *deepfakes* são resultado do uso de duas técnicas de geração: DiT-XL/2 (PEEBLES; XIE, 2023) e StyleGAN3 (KARRAS et al., 2021). Todas as amostras são precedentes de Yan et al. (2024).

Como ilustrado na Tabela 4.6, a avaliação no conjunto de dados externo DF-40, mediante imagens sintetizadas via DiT-XL/2 revelou uma queda geral no desempenho de todos os modelos em relação ao desempenho visto nos Experimentos 1 e 2, o que evidencia os desafios de generalização na detecção de *deepfakes*. Assim como exibido na Figura 4.9, a arquitetura ResNet-152 obteve o melhor desempenho relativo, atingindo 61,75% de acurácia, superando Inception-v3 (56,04%) e VGG-16 (51,69%).

A avaliação isolada da acurácia é insuficiente para compreender o comportamento dos modelos no novo domínio. Dito isso, a Tabela 4.7 organiza as métricas precisão, revocação e F1-Score segundo cada classe para melhor compreensão. A análise de revocação revela três

Tabela 4.6: Desempenho geral das arquiteturas na validação externa mediante *deepfakes* sintetizados via DiT-XL/2

Modelo	Acurácia (%)	Perda
VGG-16	51.69	2.44
ResNet-152	61.75	1.18
Inception-v3	56.04	2.19

Figura 4.9: Acurácia na validação externa mediante *deepfakes* sintetizados via DiT-XL/2

comportamentos distintos perante os dados desconhecidos: Inception-v3 detectou 74.1% dos *deepfakes*, mas penalizou fortemente as imagens autênticas (acertando apenas 36.8%); VGG-16 apresentou viés oposto, classificando a maioria das imagens como verdadeiras (65.7% de revocação para exemplos reais), deixando passar mais de 60% dos *deepfakes*; ResNet152 provou ser a arquitetura mais equilibrada e robusta nessa validação externa. Além de ter a maior acurácia geral, suas taxas de revocação foram simétricas (62.7% para *deepfakes* e 60.6% para reais).

Tabela 4.7: Métricas detalhadas por classe na validação externa mediante *deepfakes* sintetizados via DiT-XL/2.

Modelo	Classe	Precisão (%)	Revocação (%)	F1-Score (%)
VGG-16	<i>Fake</i>	54.29	38.41	44.99
	Real	50.21	65.75	56.94
ResNet-152	<i>Fake</i>	62.82	62.78	62.80
	Real	60.62	60.66	60.64
Inception-v3	<i>Fake</i>	55.43	74.12	63.43
	Real	57.38	36.89	44.90

No segundo conjunto de testes no escopo da validação externa as amostras do tipo *deepfake*

foram criadas usando a técnica StyleGAN3. Os dados consolidados na Tabela 4.8 demonstram que ResNet-152 é uma arquitetura robusta perante a tarefa, atingindo 77.34% de acurácia e a menor taxa de perda (0.65). A rede conseguiu identificar 93.10% de todos os *deepfakes*.

Tabela 4.8: Desempenho geral das arquiteturas na validação externa mediante *deepfakes* sintetizados via StyleGAN3.

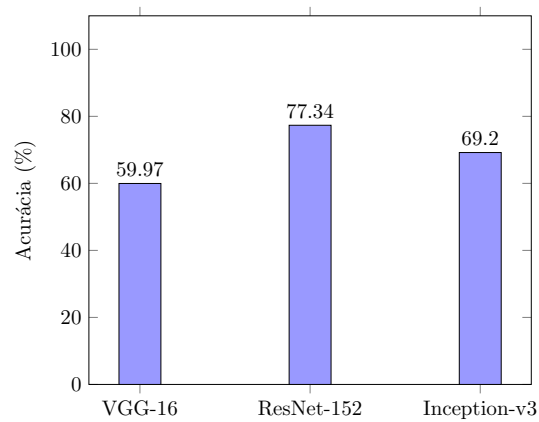
Modelo	Acurácia (%)	Perda (Loss)
VGG-16	59.97	1.49
ResNet-152	77.34	0.65
Inception-v3	69.20	1.61

A Tabela 4.9 revela um comportamento a ser destacado na Inception-v3, que obteve 69.2% de acurácia global, conforme Figura 4.10. A rede alcançou quase 100% de detecção de imagens *fake* (99.72%). Contudo, isto não reflete um aprendizado inteligente, pois apenas acertou 36.88% das faces reais. A CNN VGG-16, apesar de ter o pior desempenho global neste subconjunto (59.97% de acurácia), foi o modelo que não sofreu de um viés para apenas uma das classes, mantendo um equilíbrio de 54.52% de para falsos e 65.75% para reais, no que se refere a revocação.

Tabela 4.9: Métricas detalhadas por classe na validação externa mediante *deepfakes* sintetizados via StyleGAN3.

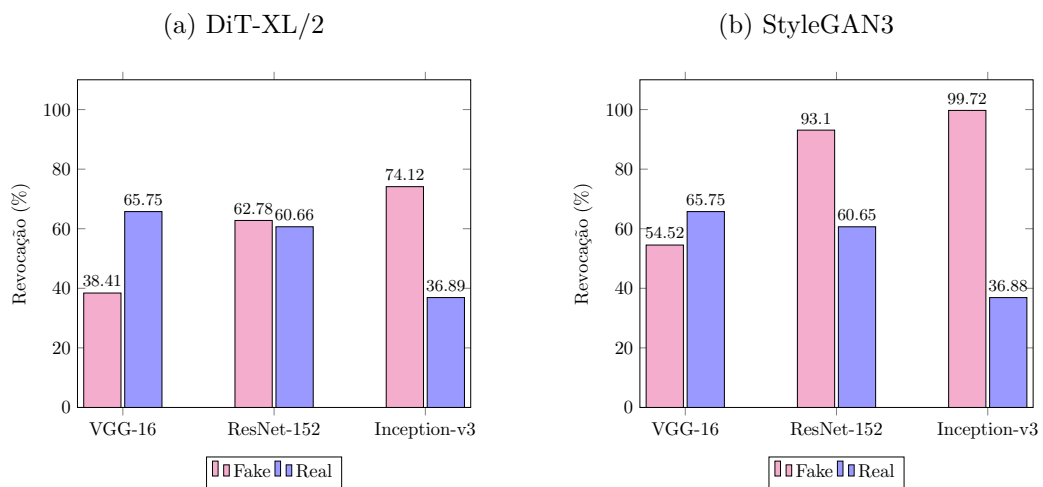
Modelo	Classe	Precisão (%)	Revocação (%)	F1-Score (%)
VGG-16	<i>Fake</i>	62.76	54.52	58.35
	Real	57.72	65.75	61.47
ResNet-152	<i>Fake</i>	71.47	93.10	80.86
	Real	89.25	60.65	72.22
Inception-v3	<i>Fake</i>	62.58	99.72	76.90
	Real	99.21	36.88	53.77

Como já esperado, os resultados no que diz respeito a classe de exemplos reais permaneceu inalterado em ambas as avaliações, com resultados de revocação constantes de 60.65%, 36.88% e 65.75% para ResNet-152, Inception-v3 e VGG-16, respectivamente. Esta constância atesta que a variação significativa observada na acurácia global provém exclusivamente do domínio das amostras sintetizadas, vide Figura 4.11. Ao manter a classe autêntica como uma variável de controle estática, torna-se evidente que a arquitetura ResNet-152 possui uma representação latente mais robusta para a adaptação e detecção de diferentes metodologias de geração

Figura 4.10: Acurácia na validação externa mediante *deepfakes* sintetizados via StyleGAN3

de *deepfakes*, em oposição à Inception-v3, dada a quantidade de falsos positivos no segundo conjunto.

Figura 4.11: Revocação por modelo na validação externa por modelo e conjunto



Capítulo 5

Considerações Finais

O presente trabalho de conclusão de curso mostrou a viabilidade dos modelos de classificação baseados em Redes Neurais Convolucionais profundas, mediante Transferência de Aprendizado, no tocante à tarefa de detecção de *deepfakes* faciais em imagens. Esses resultados foram observados a partir de experimentos computacionais com os modelos VGG-16, ResNet-152 e Inception-v3 em uma base de dados realista, abrangendo tanto a extração de características (Experimento 1) quanto o ajuste fino ou *fine-tuning* (Experimento 2). A abordagem utilizada mostrou-se capaz de identificar representações sintéticas altamente verossímeis, o que constitui um esforço importante no combate à desinformação decorrente do uso malicioso da IA Generativa.

Os resultados evidenciaram a arquitetura ResNet-152 como tendo o melhor desempenho global nos experimentos realizados, alcançando acurácia superior a 99% após ajuste fino. No entanto, uma etapa de validação externa revelou uma decréscimo geral no desempenho das arquiteturas e impuseram o desafio de generalização. Esse cenário demonstra a necessidade de ponderar que os modelos precisam conhecer mais profundamente a diversidade dos dados sintetizados em vista da rápida evolução das tecnologias para geração de *deepfakes*. Dito isso, não é possível assegurar a superioridade ou estabilidade do modelo e é necessário ainda balancear tanto os recursos computacionais quanto a capacidade adaptativa dos modelos frente ao dinamismo de novas formas de manipulação com *deepfakes*.

Em trabalhos futuros, diante dos desafios de generalização evidenciados na validação ex-

terna, almeja-se expandir o escopo de treinamento e avaliação para englobar uma diversidade maior de dados sintetizados. Junto a isso, pretende-se explorar abordagens de adaptação de domínio, buscando identificar e extrair características robustas para a identificação de *deepfakes* e mitigar o viés dos modelos aos dados de treinamento de origem.

Referências Bibliográficas

- AYDIN, Ö.; KARAARSLAN, E. Is chatgpt leading generative ai? what is beyond expectations? *Academic Platform Journal of Engineering and Smart Systems*, Akademik Perspektif Derneği, v. 11, n. 3, p. 118–134, 2023.
- BISHOP, C. M. Neural networks and their applications. *Review of scientific instruments*, American Institute of Physics, v. 65, n. 6, p. 1803–1832, 1994.
- BISHOP, C. M. *Neural networks for pattern recognition*. [S.l.]: Oxford university press, 1995.
- BRAGA, A. d. P.; LUDERMIR, T. B.; CARVALHO, A. C. P. d. L. F. d. Redes neurais artificiais: teoria e aplicações. 2000.
- BRASIL. Ministério da Ciência, Tecnologia e Inovação. *Plano Brasileiro de Inteligência Artificial (PBIA) 2024–2028: Inteligência Artificial para o Bem de Todos*. Brasília, DF, 2024. Disponível em: <<https://www.gov.br/mcti/pt-br/acompanhe-o-mcti/pbia/pbia-documento-oficial.pdf>>. Acesso em: 26 mai. 2026.
- BRYNJOLFSSON, E.; LI, D.; RAYMOND, L. Generative ai at work. *The Quarterly Journal of Economics*, Oxford University Press, v. 140, n. 2, p. 889–942, 2025.
- CHEN, Y. et al. X2-dfd: A framework for explainable and extendable deepfake detection. *Advances in Neural Information Processing Systems*, v. 38, p. 83792–83839, 2026.
- CHENG, H. et al. Diffusion facial forgery detection. In: *Proceedings of the 32nd ACM international conference on multimedia*. [S.l.: s.n.], 2024. p. 5939–5948.
- CHOLLET, F. *Deep Learning with Python*. 2. ed. Estados Unidos: Manning Publications, 2021.
- EVERINGHAM, M. et al. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, Springer, v. 111, n. 1, p. 98–136, 2015.
- GOODFELLOW, I. et al. *Deep learning*. [S.l.]: MIT press Cambridge, 2016. v. 1.
- GRIFFIN, G.; HOLUB, A.; PERONA, P. Caltech-256 object category dataset. 2007.
- HAYKIN, S. *Neural Networks and Learning Machines*. 3. ed. Estados Unidos: Pearson Prentice-Hall, 2009.
- HE, K. et al. Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2016. p. 770–778.

- JAIDKA, K. et al. Misinformation, disinformation, and generative ai: Implications for perception and policy. *Digital Government: Research and Practice*, ACM New York, NY, v. 6, n. 1, p. 1–15, 2025.
- JORDAN, M. I.; MITCHELL, T. M. Machine learning: Trends, perspectives, and prospects. *Science*, American Association for the Advancement of Science, v. 349, n. 6245, p. 255–260, 2015.
- KARRAS, T. et al. Alias-free generative adversarial networks. *Advances in neural information processing systems*, v. 34, p. 852–863, 2021.
- KHAN, S. et al. *A Guide to Convolutional Neural Networks for Computer Vision*. 1a. ed. San Rafael, California, USA: Morgan & Claypool, 2018. (Synthesis Lectures on Computer Vision).
- KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, v. 25, 2012.
- LIN, M.; CHEN, Q.; YAN, S. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.
- LIN, T.-Y. et al. Microsoft coco: Common objects in context. In: SPRINGER. *European conference on computer vision*. [S.l.], 2014. p. 740–755.
- MA, L. et al. From specificity to generality: Revisiting generalizable artifacts in detecting face deepfakes. *Advances in Neural Information Processing Systems*, v. 38, p. 69306–69344, 2026.
- MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, Springer, v. 5, n. 4, p. 115–133, 1943.
- MIRSKY, Y.; LEE, W. The creation and detection of deepfakes: A survey. *ACM computing surveys (CSUR)*, ACM New York, NY, USA, v. 54, n. 1, p. 1–41, 2021.
- MITCHELL, T. *Machine Learning*. McGraw-Hill Education, 1997. (McGraw-Hill international editions - computer science series). ISBN 9780070428072. Disponível em: <<https://books.google.com.br/books?id=xOGAngEACAAJ>>.
- NAFFI, N. *Deepfakes and the crisis of knowing*. 2025. Disponível em: <<https://www.unesco.org/en/articles/deepfakes-and-crisis-knowing>>.
- PAN, S. J.; YANG, Q. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, IEEE, v. 22, n. 10, p. 1345–1359, 2009.
- PARK, S.; NAN, X. Generative ai and misinformation: a scoping review of the role of generative ai in the generation, detection, mitigation, and impact of misinformation. *AI & SOCIETY*, Springer, p. 1–15, 2025.
- PEEBLES, W.; XIE, S. Scalable diffusion models with transformers. In: *Proceedings of the IEEE/CVF international conference on computer vision*. [S.l.: s.n.], 2023. p. 4195–4205.
- RANA, M. S. et al. Deepfake detection: A systematic literature review. *IEEE access*, IEEE, v. 10, p. 25494–25513, 2022.
- RICH, E.; KNIGHT, K. et al. *Inteligencia artificial*. Madrid: McGraw-Hill,, 1994.

- ROSSLER, A. et al. Faceforensics++: Learning to detect manipulated facial images. In: *Proceedings of the IEEE/CVF international conference on computer vision*. [S.l.: s.n.], 2019. p. 1–11.
- RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. *nature*, Nature Publishing Group UK London, v. 323, n. 6088, p. 533–536, 1986.
- RUSSAKOVSKY, O. et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, Springer, v. 115, n. 3, p. 211–252, 2015.
- RUSSELL, S.; NORVIG, P. *Artificial Intelligence – A Modern Approach*. 3. ed. Nova Jersey: Pearson, 2016.
- SANCHES, D.; MENEZES, P. *Conectividade significativa e desinformação: quais são as competências e os saberes necessários na era da IA generativa?* 2025. In: NIC.br. **Pesquisa sobre o uso das tecnologias de informação e comunicação nos domicílios brasileiros: TIC Domicílios 2024**. São Paulo: CGI.br, pp. 93–101. Capítulo de Livro. Acesso em: 18 jan. 2026.
- SARKAR, D.; BALI, R.; GHOSH, T. *Hands-On Transfer Learning with Python: Implement advanced deep learning and neural network models using TensorFlow and Keras*. [S.l.]: Packt Publishing Ltd, 2018.
- SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- SZEGEDY, C. et al. Going deeper with convolutions. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2015. p. 1–9.
- SZEGEDY, C. et al. Rethinking the inception architecture for computer vision. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2016. p. 2818–2826.
- WANG, G. et al. Fsfm: A generalizable face security foundation model via self-supervised facial representation learning. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. [S.l.: s.n.], 2025. p. 24364–24376.
- WIDROW, B.; HOFF, M. E. Adaptive switching circuits. In: *Neurocomputing: foundations of research*. [S.l.: s.n.], 1988. p. 123–134.
- World Economic Forum. *The Global Risks Report 2025*. Geneva, Switzerland, 2025. Disponível em: <<https://www.weforum.org/publications/global-risks-report-2025/>>.
- XIE, S. et al. Deepfake detection in the aigc era: A survey, benchmarks, and future perspectives. *Information Fusion*, Elsevier, p. 103740, 2025.
- YAN, Z. et al. Df40: Toward next-generation deepfake detection. *Advances in Neural Information Processing Systems*, v. 37, p. 29387–29434, 2024.
- ZHANG, A. et al. *Dive into Deep Learning*. Cambridge University Press, 2023. ISBN 9781009389433. Disponível em: <<https://books.google.com.br/books?id=vfDiEAAAQBAJ>>.

ZHAO, X. et al. A review of convolutional neural networks in computer vision. *Artificial Intelligence Review*, v. 57, n. 4, 2024.