

# Comparação de CNNs para Detecção de *DeepFakes*

Um estudo comparativo entre CNNs na tarefa de classificação binária de imagens falsas e reais

Thiago de Queiroz Osório  
PPG – Ciência da Computação  
UNIFESP  
São José dos Campos, SP  
thiago.q.osorio@gmail.com

**Abstract**—This paper presents a comparative analysis of various convolutional neural networks (CNN) architectures applied to deepfake detection. Using a public dataset containing real and fake images, three main strategies were evaluated: manual implementation of classical CNN architectures (LeNet-5, AlexNet, Inception-v1); transfer learning with lightweight networks (MobileNet v1, EfficientNetB0, DenseNet121); and transfer learning with traditional deep architectures (ResNet50, VGG-16, Xception). VGG-16 achieved the best results on the primary test set, with 95% accuracy, 97% precision, and 93% recall. However, when tested on an external dataset composed of synthetic images from a different domain, its performance dropped to 55% accuracy, revealing limitations in the model's generalization capability. These findings underscore the effectiveness of transfer learning while highlighting the need for additional strategies to address cross-domain variability.

**Keywords**—*Deepfake, Computer Vision, Convolutional Neural Networks, Transfer Learning, Model Generalization.*

**Resumo**— Este artigo realiza uma análise comparativa de diferentes arquiteturas de redes neurais convolucionais (CNNs) na tarefa de detecção de deepfakes. Utilizando uma base de dados pública composta por imagens reais e falsas, foram testadas três estratégias principais: replicação manual de arquiteturas clássicas (LeNet-5, AlexNet, Inception-v1); uso de transfer learning com redes leves (MobileNet v1, EfficientNetB0, DenseNet121); e uso de transfer learning com redes profundas tradicionais (ResNet50, VGG-16, Xception). A VGG-16 obteve os melhores resultados no conjunto de teste principal, com acurácia de 95%, precisão de 97% e sensibilidade de 93%. No entanto, ao ser avaliada em uma base externa composta por imagens sintéticas de outro domínio, seu desempenho caiu para 55% de acurácia, evidenciando limitações na capacidade de generalização do modelo. Os resultados destacam a eficácia do transfer learning, mas também apontam a necessidade de estratégias adicionais para lidar com variabilidade entre domínios.

**Palavras-chave**—*Deepfake, Visão Computacional, Redes Neurais Convolucionais, Transfer Learning, Generalização de Modelo.*

## I. INTRODUÇÃO

Com o avanço de técnicas de geração de conteúdo sintético baseadas em inteligência artificial, especialmente as Redes Geradoras Adversariais (GANs), tornou-se cada vez mais difícil distinguir imagens reais de falsas. Esse fenômeno deu origem aos chamados *deepfakes*, que têm sido usados de forma crescente para fins maliciosos, como fraudes financeiras, manipulação política e disseminação de desinformação [10], [11], [12]. A sofisticação desses conteúdos exige soluções automatizadas e robustas para sua detecção, o que tornou a detecção de *deepfakes* um dos principais desafios da Visão Computacional moderna.

Nesse cenário, redes neurais convolucionais (CNNs) destacam-se como ferramentas poderosas para tarefas de classificação de imagens, por sua capacidade de aprender representações hierárquicas e invariantes a pequenas variações. Desde a introdução da LeNet-5, que marcou um ponto de virada no reconhecimento de caracteres manuscritos [1], passando pelo impacto da AlexNet no ImageNet [2] e o uso de arquiteturas profundas como Inception-v1 [3], as CNNs têm sido constantemente aprimoradas para oferecer maior profundidade, eficiência e capacidade de generalização.

Mais recentemente, surgiram arquiteturas otimizadas para dispositivos com recursos limitados, como a MobileNet v1 [4], a EfficientNetB0 [5] e a DenseNet121 [6], que combinam desempenho competitivo com menor custo computacional. Paralelamente, redes mais profundas como ResNet50 [7], VGG-16 [8] e Xception [9] continuam sendo amplamente utilizadas em cenários onde a acurácia é prioritária.

Este trabalho propõe uma análise comparativa entre diferentes abordagens baseadas em CNNs para detecção de imagens *deepfake*. Utilizando uma base pública do Kaggle composta por imagens reais e sintéticas [*Deepfake and Real Images*, Kaggle], testamos três estratégias distintas: (i) implementação manual de arquiteturas clássicas (LeNet-5, AlexNet e Inception-v1), (ii) uso de *transfer learning* com redes leves (MobileNet v1, EfficientNetB0, DenseNet121) e (iii) uso de *transfer learning* com redes profundas tradicionais (ResNet50, VGG-16 e Xception), conectadas a uma camada final para previsão da probabilidade da imagem ser falsa.

O objetivo deste estudo é investigar o desempenho dessas diferentes abordagens na tarefa de classificação binária (real ou fake), analisando acurácia, precisão, sensibilidade, estabilidade e capacidade de generalização. Acredita-se que tal comparação empírica possa contribuir para a seleção de arquiteturas mais adequadas em cenários reais de combate a *deepfakes*, inclusive em contextos sensíveis como redes sociais, sistemas bancários e mecanismos de autenticação facial [10], [11].

O restante deste artigo está organizado da seguinte forma: a Seção II apresenta os principais trabalhos relacionados; a Seção III descreve a metodologia adotada; a Seção IV traz os resultados obtidos e suas análises; e a Seção V apresenta as conclusões e perspectivas futuras.

## II. TRABALHOS RELACIONADOS

A detecção de deepfakes é uma tarefa desafiadora, que envolve a identificação de padrões sutis em imagens geradas artificialmente por redes generativas adversariais (GANs). Para isso, redes neurais convolucionais (CNNs) vêm sendo amplamente utilizadas, com diversos estudos propondo arquiteturas e estratégias para melhorar a acurácia e a robustez frente às manipulações sintéticas.

A LeNet-5 foi uma das primeiras redes convolucionais eficazes, proposta por LeCun et al. [1] para o reconhecimento automático de dígitos manuscritos. Sua arquitetura, embora simples, introduziu conceitos fundamentais como camadas convolucionais seguidas de *pooling* e camadas totalmente conectadas, sendo uma referência histórica.

Posteriormente, a AlexNet [2] demonstrou o poder das CNNs profundas ao vencer o *ImageNet Large Scale Visual Recognition Challenge (ILSVRC)* em 2012, popularizando o uso de GPUs para treinamento de redes profundas. Já a Inception-v1, proposta por Szegedy et al. [3], introduziu o conceito de *Inception modules*, permitindo redes mais profundas e eficientes por meio de convoluções com múltiplos tamanhos de filtro dentro do mesmo bloco.

Essas redes formam a base da evolução das CNNs modernas e continuam sendo utilizadas como ponto de partida ou *benchmarking* para novas propostas.

## B. Arquiteturas modernas otimizadas

Com o objetivo de reduzir o custo computacional sem comprometer a performance, surgiram arquiteturas mais leves e otimizadas. A MobileNet v1 [4] utiliza convoluções separáveis em profundidade (*depthwise separable convolutions*) para reduzir o número de parâmetros, tornando-a adequada para aplicações móveis e embarcadas.

A EfficientNet [5] propõe uma abordagem sistemática de escalonamento da arquitetura (em largura, profundidade e resolução), obtendo redes eficientes e altamente acuradas com menos recursos. Por sua vez, a DenseNet [6] introduz conexões densas entre todas as camadas de uma mesma etapa, facilitando o fluxo de gradientes e promovendo reutilização de características.

As redes ResNet [7] e VGG-16 [8] também se destacam como modelos base para *transfer learning*. A ResNet introduz conexões de atalho (*skip connections*), solucionando o problema de degradação em redes profundas, enquanto a VGG-16 é reconhecida por sua simplicidade e profundidade com uso de filtros 3×3 empilhados.

A arquitetura Xception [9], por fim, combina a ideia de separação espacial e em profundidade, sendo uma generalização eficiente da *Inception* com desempenho competitivo em diversas tarefas.

## C. Detecção de deepfakes

Com o aumento do uso de mídias sintéticas, a detecção de *deepfakes* tornou-se um campo de pesquisa em rápida expansão. Em [10], os autores discutem os riscos associados às fraudes baseadas em *deepfakes* e ressaltam a importância de mecanismos de detecção confiáveis para garantir a integridade digital.

No estudo de Hossain et al. [11], propõe-se o uso de modelos baseados em GANs para detectar conteúdos manipulados, com foco em aplicações financeiras e transações online. Já em [12], é apresentada uma revisão dos principais desafios e tendências relacionadas à disseminação de *deepfakes* em redes sociais, evidenciando o papel crítico de classificadores automáticos na mitigação desses riscos.

Estes trabalhos reforçam a relevância de estudos como o presente, que investigam empiricamente quais arquiteturas de CNN apresentam melhor desempenho na detecção de imagens falsas, considerando diferentes estratégias de aprendizado e complexidade computacional.

Esta seção descreve as etapas realizadas para avaliar diferentes arquiteturas de redes neurais convolucionais (CNNs) na detecção de imagens *deepfake*. O estudo envolveu: (i) preparação dos dados, (ii) definição das arquiteturas testadas, (iii) treinamento e avaliação inicial, e (iv) teste de generalização em uma base de dados externa.

### A. Conjuntos de Dados

#### 1) Base primária (treinamento e validação)

A primeira base utilizada foi obtida do Kaggle [13], contendo imagens classificadas como reais ou falsas (*deepfakes*). Os dados foram organizados em três subconjuntos:

- Treinamento: ~73%
- Validação: ~21%
- Teste: ~6%

As imagens foram redimensionadas para 128x128 pixels e normalizadas no intervalo [0,1] utilizando a função *normalizer*, que converte os dados para o tipo *float32* e realiza divisão por 255. Os *datasets* foram carregados com *image\_dataset\_from\_directory*, com *batch size* igual a 32.

#### 2) Base de generalização

Para avaliar a capacidade de generalização dos modelos, foi utilizada uma segunda base externa [14], também do Kaggle, contendo 140 mil imagens balanceadas entre rostos reais e rostos gerados por GANs. Esta base foi mantida completamente isolada do processo de treinamento, sendo utilizada apenas na fase final do estudo. O mesmo processo de redimensionamento e normalização foi aplicado para garantir consistência nos testes.

### B. Arquiteturas Avaliadas

#### 1) Redes implementadas manualmente

Foram implementadas em TensorFlow/Keras três arquiteturas clássicas, com base em suas descrições originais:

- LeNet-5: composta por duas camadas convolucionais com ativação *tanh*, seguidas por *average pooling*, e três camadas densas, finalizando com uma camada sigmoide para saída binária.
- AlexNet: rede profunda com cinco camadas convolucionais, *ReLU*, *MaxPooling* e duas camadas densas de 4096 neurônios, com *Dropout* para regularização.
- Inception-v1: construída manualmente com *Inception Modules*, combinando convoluções 1x1, 3x3, 5x5 e *pooling projections*, seguida de *AveragePooling*, *Flatten* e uma camada densa final.

Todas as redes utilizaram como entrada imagens RGB de tamanho 128x128x3.

### C. Treinamento dos Modelos

Os modelos foram treinados utilizando a função de perda *binary\_crossentropy*, com otimizador Adam e taxa de aprendizado fixada em 0.0001. O número de épocas foi limitado a 10, e os modelos foram validados a cada época no conjunto de validação.

Durante o treinamento, foram monitoradas as seguintes métricas:

- Acurácia
- Precisão
- Sensibilidade (Sensibilidade)

A implementação utilizou *model.fit()* para o treinamento e *model.evaluate()* para avaliação no conjunto de teste, capturando as métricas diretamente do Keras. O *callback* de *early stopping* pode ser adicionado em futuras execuções para evitar *overfitting*, embora não tenha sido utilizado nesta fase.

#### D. Avaliação e Métricas

Os modelos foram avaliados no conjunto de teste da base original e, posteriormente, no conjunto externo de generalização. As seguintes métricas foram consideradas:

- Acurácia: proporção de predições corretas sobre o total de amostras.
- Precisão: razão entre verdadeiros positivos e o total de predições positivas.
- Sensibilidade (Sensibilidade): razão entre verdadeiros positivos e o total de exemplos realmente positivos.

Essas métricas permitem avaliar o modelo de forma abrangente, considerando especialmente os falsos negativos e falsos positivos (que são críticos em sistemas de detecção de fraudes).

#### E. Etapa de Generalização

Após o treinamento e a avaliação em teste, a arquitetura com melhor desempenho foi selecionada para a etapa de teste de generalização. Nessa fase, o modelo foi aplicado diretamente na base *140k Real and Fake Faces* [14], sem re-treinamento ou ajuste de pesos. O objetivo foi verificar se o modelo treinado em uma base específica seria capaz de generalizar para imagens geradas por técnicas distintas, simulando um cenário real de detecção de *deepfakes* "fora da amostra".

A avaliação nessa etapa seguiu o mesmo conjunto de métricas descritas anteriormente, permitindo comparações diretas de desempenho entre domínios distintos.

#### F. Ferramentas Utilizadas

Todos os experimentos foram implementados em Python 3.9 utilizando TensorFlow/Keras. Os testes foram conduzidos em uma instância do SageMaker. O código-fonte está disponível publicamente no repositório do autor: [https://github.com/thiago-osorio/mestrado/tree/main/visao\\_computacional](https://github.com/thiago-osorio/mestrado/tree/main/visao_computacional).

### IV. ANÁLISE EXPERIMENTAL

Nesta seção, são apresentados e discutidos os resultados obtidos para cada arquitetura de rede neural convolucional testada. As métricas consideradas foram: acurácia, precisão e sensibilidade (sensibilidade), conforme descrito na Seção III. A Tabela I resume os valores obtidos nos testes realizados com a base de dados principal.

#### A. Desempenho das Arquiteturas Replicadas

As redes LeNet-5, AlexNet e Inception-v1, implementadas manualmente a partir de suas descrições originais, apresentaram desempenhos variados, refletindo tanto suas limitações arquiteturais quanto sua adequação (ou não) ao contexto atual de detecção de imagens sintéticas.

- LeNet-5 obteve acurácia de 77%, precisão de 79% e sensibilidade de 73%. Embora esses valores sejam aceitáveis, sua arquitetura superficial, originalmente desenhada para tarefas simples, limitou sua capacidade de extrair representações discriminativas em imagens de maior complexidade visual, como é o caso de *deepfakes*.

- AlexNet teve desempenho superior, com acurácia de 87%, precisão de 92% e sensibilidade de 81%, beneficiando-se de sua profundidade e da presença de mecanismos como *ReLU* e *dropout*. Sua capacidade de capturar padrões mais sutis contribuiu para uma detecção mais eficaz de falsificações.
- Inception-v1, por outro lado, apresentou comportamento anômalo: acurácia de 50%, precisão de 50% e sensibilidade de 100%. Este resultado indica que a rede classificou todas as imagens como falsas (classe positiva), o que aponta para uma falha na convergência ou uma forte tendência ao viés. A complexidade estrutural da arquitetura pode ter exigido ajustes adicionais não aplicados neste experimento.

#### B. Desempenho das Redes Leves (Lightweight CNNs)

Entre as arquiteturas otimizadas para eficiência, os resultados também foram diversos:

- MobileNet v1 atingiu acurácia de 87%, com precisão de 94% e sensibilidade de 78%, demonstrando ser uma solução leve e eficaz. Sua estrutura baseada em convoluções separáveis em profundidade proporcionou boa capacidade de generalização com baixa complexidade computacional.
- EfficientNetB0 apresentou resultados altamente desequilibrados: acurácia de 51%, precisão de 99% e sensibilidade de apenas 1%. Essa disparidade sugere que o modelo aprendeu a prever majoritariamente a classe negativa (imagens reais), falhando em detectar *deepfakes* (o que é problemático em cenários sensíveis).
- DenseNet121 teve excelente desempenho: acurácia de 91%, precisão de 98% e sensibilidade de 83%, combinando alta precisão com boa capacidade de identificação de falsificações. As conexões densas entre camadas contribuíram para a reutilização de características e redução do *overfitting*.

#### C. Desempenho com Redes Profundas via Transfer Learning

As redes com melhores resultados foram obtidas via *transfer learning*, especialmente com arquiteturas profundas já consolidadas:

- ResNet50 apresentou acurácia, precisão e sensibilidade de 86%, um desempenho equilibrado graças às conexões residuais que facilitam o treinamento de redes profundas.
- Xception também teve bom desempenho (acurácia de 88%, precisão de 91%, sensibilidade de 84%), evidenciando a eficácia das convoluções separáveis em profundidade na extração de padrões locais relevantes.
- VGG-16, por sua vez, obteve os melhores resultados globais com acurácia de 95%, precisão de 97% e sensibilidade de 93%. Mesmo sendo uma arquitetura mais antiga, sua estrutura sequencial e estável contribuiu para uma aprendizagem eficaz. A VGG-16 se destacou não apenas pela performance, mas também pela consistência das métricas, sendo escolhida como a rede com melhor desempenho geral nesta etapa.

#### D. Avaliação de Generalização

Para avaliar a robustez do modelo selecionado frente a dados desconhecidos, a VGG-16 foi submetida a uma nova base de dados contendo 140 mil imagens reais e falsas

geradas por GANs [14]. Essa base não foi utilizada durante o treinamento, o que permitiu avaliar a capacidade de generalização do modelo para domínios diferentes.

O desempenho da VGG-16 caiu significativamente nesta etapa:

- Acurácia: 55%
- Precisão: 55%
- Sensibilidade: 62%

Essa queda substancial revela um desafio importante de generalização: embora o modelo tenha aprendido a identificar padrões presentes na base original, ele teve dificuldade em lidar com imagens geradas por outras técnicas ou com características diferentes daquelas vistas durante o treinamento.

A precisão de 55% indica que o modelo gerou um número elevado de falsos positivos, enquanto a sensibilidade de 62% mostra que ele ainda foi capaz de identificar parte das imagens falsas corretamente, embora com eficácia reduzida. Isso pode ser atribuído a diferenças no domínio visual entre as duas bases, por exemplo: variações na resolução; estilo de geração; pós-processamento; tipo de GAN utilizado.

Esses resultados evidenciam a importância de incluir dados diversos e heterogêneos durante o treinamento, bem como de realizar testes sistemáticos de generalização antes de implantar modelos em ambientes reais. Uma possível linha futura seria aplicar técnicas de *ensemble learning* para melhorar a robustez do sistema frente a dados fora da distribuição original.

## V. CONCLUSÃO

Este trabalho apresentou uma análise comparativa de diferentes arquiteturas de redes neurais convolucionais aplicadas à tarefa de detecção de *deepfakes*, utilizando uma base de imagens do *Kaggle* rotulada como real ou falsa. Foram testadas arquiteturas clássicas replicadas manualmente (LeNet-5, AlexNet e Inception-v1), redes leves modernas com transfer learning (MobileNet v1, EfficientNetB0, DenseNet121) e arquiteturas profundas consagradas (ResNet50, VGG-16 e Xception).

Os resultados demonstraram que, embora modelos clássicos como LeNet-5 e AlexNet possam apresentar desempenho razoável, as melhores performances foram alcançadas com arquiteturas modernas por meio de *transfer learning*. A VGG-16 destacou-se com os melhores indicadores de desempenho no conjunto de teste principal, atingindo acurácia de 95%, precisão de 97% e sensibilidade de 93%. Esse resultado mostra que, mesmo sendo uma arquitetura relativamente antiga, sua estrutura profunda e regular é altamente eficaz em tarefas de classificação binária como a detecção de *deepfakes*.

Contudo, ao ser avaliada em uma base de dados externa, composta por imagens falsas geradas por técnicas distintas de GAN, a VGG-16 apresentou uma queda significativa de desempenho (acurácia de 55%, precisão de 55%, recall de 62%), evidenciando a dificuldade do modelo em generalizar para outros domínios de *deepfakes*. Essa limitação aponta para a necessidade de desenvolver modelos mais robustos, treinados com dados mais diversos ou com estratégias mais específicas.

Os principais aprendizados deste trabalho podem ser resumidos em três pontos:

- A curadoria da base de dados é crítica: modelos que se saem bem em bases específicas podem falhar

drasticamente ao enfrentar dados com características diferentes.

- *Transfer learning* é uma estratégia eficaz, especialmente com arquiteturas bem estabelecidas como VGG-16 e DenseNet121, que mostraram desempenho equilibrado entre precisão e sensibilidade.
- Avaliações de generalização são fundamentais para validar o modelo em cenários do mundo real (um aspecto muitas vezes negligenciado em estudos acadêmicos).

Como trabalhos futuros, recomenda-se:

- Expandir o treinamento para incluir múltiplas bases de dados com diferentes tipos de manipulações;
- Aplicar técnicas de *data augmentation* e *fine-tuning* mais agressivas para melhorar a generalização;
- Explorar métodos de *ensemble learning* para lidar com variabilidade inter-base;
- Avaliar o uso de arquiteturas híbridas ou especializadas para detecção de artefatos em diferentes escalas e domínios.

Em suma, este estudo reforça a relevância das CNNs como ferramenta eficaz para detecção de *deepfakes*, mas também evidencia os desafios reais associados à generalização e robustez dos modelos em ambientes fora da distribuição de treinamento.

## REFERÊNCIAS BIBLIOGRÁFICAS

1. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-Based Learning Applied to Document Recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998. [Online]. Available: <https://ieeexplore.ieee.org/document/726791>
2. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems*, 2012. [Online]. Available: [https://papers.nips.cc/paper\\_files/paper/2012/hash/c399862d3b9d6b76c8436e924a68e45b-Abstract.html](https://papers.nips.cc/paper_files/paper/2012/hash/c399862d3b9d6b76c8436e924a68e45b-Abstract.html)
3. C. Szegedy et al., "Going Deeper with Convolutions," *arXiv preprint*, arXiv:1409.4842, 2014. [Online]. Available: <https://arxiv.org/abs/1409.4842>
4. A. G. Howard et al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *arXiv preprint*, arXiv:1704.04861, 2017. [Online]. Available: <https://arxiv.org/abs/1704.04861>
5. M. Tan and Q. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," *arXiv preprint*, arXiv:1905.11946, 2019. [Online]. Available: <https://arxiv.org/abs/1905.11946>
6. G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," *arXiv preprint*, arXiv:1608.06993, 2016. [Online]. Available: <https://arxiv.org/abs/1608.06993>
7. K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *arXiv preprint*, arXiv:1512.03385, 2015. [Online]. Available: <https://arxiv.org/abs/1512.03385>
8. K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv preprint*, arXiv:1409.1556, 2014. [Online]. Available: <https://arxiv.org/abs/1409.1556>
9. F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," *arXiv preprint*, arXiv:1610.02357, 2016. [Online]. Available: <https://arxiv.org/abs/1610.02357>
10. A. Pathak, "Deepfake Fraud Detection: Safeguarding Trust in Generative AI," SSRN, May 2024. [Online]. Available: [https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=5031627](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=5031627)

11. M. R. Hossain, M. A. R. Ahad, and S. A. Mostafa, "Detection of AI Deepfake and Fraud in Online Payments Using GAN-Based Models," *arXiv preprint*, arXiv:2501.07033, 2025. [Online]. Available: <https://arxiv.org/pdf/2501.07033>
12. A. A. Khan, "Deepfake Technology: Overview and Emerging Trends in Social Media," SSRN, April 2024. [Online]. Available: [https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=4981040](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4981040)
13. M. Karki, "Deepfake and Real Images," *Kaggle*, [Online]. Available: <https://www.kaggle.com/datasets/manjilkarki/deepfake-and-real-images>
14. X. Lu, "140K Real and Fake Faces," *Kaggle*, [Online]. Available: <https://www.kaggle.com/datasets/xhlulu/140k-real-and-fake-faces>