Universidade Federal do Paraná
Setor de Ciências Exatas
Departamento de Estatística
Programa de Especialização em *Data Science* e *Big Data*

Vitor Couto de Avanço

Análise de Custo-Benefício e Desempenho de Modelos Pré-Treinados em Tarefas de Classificação de Imagens

Curitiba
2025

Vitor Couto de Avanço

Análise de Custo-Benefício e Desempenho de Modelos Pré-Treinados em Tarefas de Classificação de Imagens

Monografia apresentada ao Programa de Especialização em *Data Science* e *Big Data* da Universidade Federal do Paraná como requisito parcial para a obtenção do grau de especialista.

Orientador: Prof. Paulo R. Lisboa de Almeida

Curitiba
2025

Análise de Custo-Benefício e Desempenho de Modelos Pré-Treinados em Tarefas de Classificação de Imagens

Vitor Couto de Avanço

Aluno do programa de Especialização em Data Science & Big Data da Universidade Federal do Paraná, UFPR

Este trabalho apresenta uma investigação experimental sobre o desempenho de diferentes arquiteturas de redes neurais profundas aplicadas à classificação de imagens. Foram avaliadas as redes MobileNetV3 Small, EfficientNet-B2 e ViT-B16, combinadas com os métodos de treinamento *transfer learning* e *fine-tuning*, sobre três conjuntos de dados com características distintas: FashionMNIST, CIFAR-10 e PKLot. A partir de cinco execuções independentes por experimento, foram analisadas acurácia, estabilidade dos resultados, tempo de inferência por imagem e tempo total de treinamento. Os resultados evidenciam que, embora algumas combinações apresentem acurácias superiores em determinados contextos, não há uma solução única que se destaque de forma consistente em todos os cenários avaliados. Além disso, os experimentos demonstram que métricas preditivas como a acurácia, embora relevantes, não são suficientes para embasar a escolha de um modelo, sendo fundamental considerar os custos computacionais e operacionais envolvidos em sua aplicação real.
Palavras-chave: redes neurais, datasets, métodos de treinamento, custo computacional, classificação de imagens.

An experimental investigation into the performance of different deep neural network architectures applied to image classification tasks. The architectures MobileNetV3 Small, EfficientNet-B2, and ViT-B16 were evaluated in combination with the training methods Transfer Learning and Fine-Tuning across three datasets with distinct characteristics: FashionMNIST, CIFAR-10, and PKLot. Each experiment was independently executed five times, enabling the analysis of accuracy, result stability, inference time per image, and total training time. The results show that although some combinations achieved superior accuracy in specific contexts, no single architecture and training method consistently outperformed the others across all scenarios. Furthermore, the experiments demonstrate that predictive metrics such as accuracy, while important, are not sufficient to guide model selection, reinforcing the need to also consider computational and operational costs in real-world applications.
Keywords: neural networks, datasets, training methods, computational cost, image classification

1. Introdução

O avanço das redes neurais profundas consolidou-se como um dos principais marcos do aprendizado de máquina contemporâneo, impulsionando soluções em tarefas de classificação de imagens, detecção de objetos e processamento de linguagem natural[1, 2]. Dentre as diversas abordagens existentes, técnicas como o *transfer learning* e o *fine-tuning* destacam-se por possibilitar o aproveitamento de representações previamente aprendidas em grandes bases de dados, o que reduz significativamente a necessidade de dados rotulados e tempo de treinamento dos modelos. Ao mesmo tempo, a evolução arquitetural das redes neurais, com propostas que vão desde modelos compactos, como a MobileNet, até estruturas mais profundas e sofisticadas, como os Vision Transformers, ampliou de forma substancial as possibilidades de aplicação em diferentes domínios.

Apesar dos avanços notáveis em desempenho preditivo, a escolha da combinação mais adequada entre arquitetura e método de treinamento permanece um desafio prático e científico. Ainda que diversos estudos apresentem métricas de desempenho em *benchmarks* consolidados, muitas vezes os custos computacionais associados ao treinamento e à utilização em produção são subestimados ou negligenciados [3, 4]. Essa lacuna evidencia a necessidade de investigações experimentais que não apenas quantifiquem a acurácia dos modelos, mas também caracterizem de maneira

sistemática o custo operacional e o tempo necessário para treinar e executar diferentes redes.

Nesse contexto, este trabalho propõe uma análise experimental que compara o desempenho de três arquiteturas de redes neurais profundas: MobileNetV3 Small [8], EfficientNet-B2 [9] e Vision Transformer ViT-B16 [10]. Aplicadas a conjuntos de dados com diferentes graus de complexidade: FashionMNIST [5], CIFAR-10 [6] e PKLot [7]. São avaliados dois métodos de ajuste amplamente utilizados na literatura, *transfer learning* e *fine-tuning*, com o objetivo de identificar os impactos dessas escolhas tanto na acurácia quanto na variabilidade dos resultados, no tempo de inferência e no custo total de treinamento.

Este artigo está estruturado da seguinte maneira: a Seção 2 descreve em detalhes os conjuntos de dados e as arquiteturas das redes utilizadas, contextualizando suas principais características e complexidades. A Seção 3 apresenta o protocolo experimental adotado, incluindo os procedimentos de treinamento, critérios de parada e métricas de avaliação. A Seção 4 expõe os resultados obtidos e discute comparativamente os principais achados. Por fim, a Seção 5 apresenta as conclusões.

2. Bases de Dados e Redes Neurais

A diversidade entre conjuntos de dados e arquiteturas de redes neurais impactam diretamente a capacidade de generalização dos modelos de aprendizado. Combinando bases com diferentes níveis de complexidade e modelos com distintas arquiteturais é possível avaliar o desempenho em múltiplos contextos de classificação de imagens.

2.1. Conjuntos de Dados

Todos os conjuntos de dados utilizados são compostos por imagens rotuladas, selecionados com o intuito de representar diferentes graus de complexidade, tamanhos amostrais e desafios computacionais.

O FashionMNIST [5] é um conjunto de dados desenvolvido como uma alternativa mais desafiadora ao clássico MNIST [13]. Composto por 70.000 imagens em escala de cinza, com dimensão de 28×28 pixels, o FashionMNIST representa peças de vestuário e é dividida em 60.000 imagens para treino e 10.000 para teste. As imagens são agrupadas em 10 categorias: Camiseta, Calça, Suéter, Vestido, Casaco, Sandália, Camisa, Tênis, Bolsa e Bota. Por se tratar de imagens monocromáticas e centradas, a base é considerada relativamente simples para modelos modernos, sendo especialmente útil para estudos iniciais.

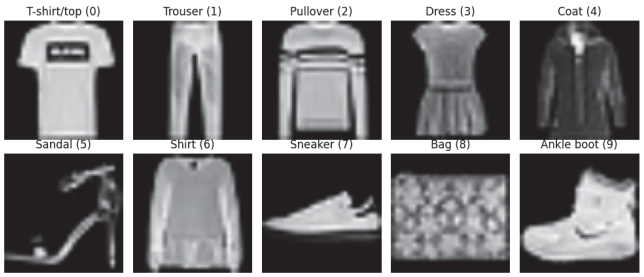

Figura 1: Classes do Dataset FashionMNIST.

O CIFAR-10 [6] é um *benchmark* clássico na literatura. Contém 60.000 imagens coloridas (32×32 pixels), organizadas em 10 classes distintas de objetos: Avião, Automóvel, Pásssaro, Gato, Cervo, Cachorro, Sapo, Cavalo, Navio e Caminhão. As imagens são divididas em 50.000 imagens para treino e 10.000 para teste. Ao contrário do FashionMNIST, os dados do CIFAR-10 apresentam maior variabilidade de formas, cores e contextos, o que exige maior capacidade de generalização dos modelos.

Figura 2: Classes do Dataset CIFAR10.

Por fim, o PKLot [7] é um conjunto de dados de imagens reais de estacionamentos, capturadas por câmeras fixas em diferentes condições climáticas e horários. Com aproximadamente 700.000 imagens segmentadas entre duas classes: "ocupado"e "vazio", é uma base consideravelmente mais complexa e extensa do que as anteriores. Além da alta variação de iluminação e perspectiva, a base exige robustez dos modelos frente a dados desbalanceados e ruídos visuais. As imagens utilizadas neste estudo foram extraídas da versão segmentada do PKLot, reorganizadas manualmente em subconjuntos de treino (37,5%), validação (12,5%) e teste (50%), com separação feita com base nas datas das capturas das imagens, de forma a preservar a integridade temporal dos lotes.

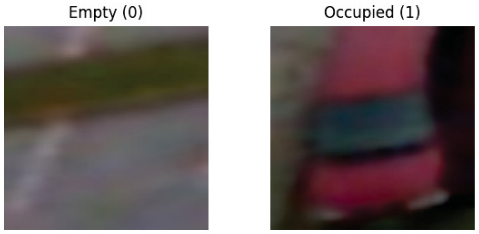

Figura 3: Classes do Dataset PKLOT.

2.2. Redes Neurais

Redes neurais artificiais são modelos computacionais inspirados na estrutura do cérebro humano, compostos por camadas de unidades chamadas *perseptrons*, organizadas de forma a permitir o aprendizado de padrões complexos a partir de dados de treinamento.

Quando organizadas em múltiplas camadas, formando as chamadas redes neurais profundas (*deep neural networks*), esses modelos tornam-se especialmente eficazes em tarefas como reconhecimento de imagens, tradução automática e processamento de linguagem natural. Seu funcionamento baseia-se em ajustar pesos entre os neurônios ao longo de um processo de treinamento, a fim de minimizar o erro entre as previsões do modelo e os resultados esperados.

Três redes neurais profundas foram selecionadas para os experimentos. Todas utilizam pesos pré-treinados do ImageNet, um vasto conjunto de dados com mais de 14 milhões de imagens, permitindo uma análise comparativa de desempenho e custo computacional.

Tabela 1: Informações das Redes Neurais utilizadas no artigo.

	Ano de Publicação	Nº de Parâmetros	Tamanho
MobileNetV3 S	2019	$2{,}5 \times 10^6$	9,8 MB
EfficientNet-B2	2019	$9{,}1 \times 10^6$	35,2 MB
ViT-B16	2020	$8{,}6 \times 10^7$	330,3 MB

As arquiteturas apresentam características distintas. A MobileNetV3 Small, é uma rede convolucional leve e eficiente, ideal para dispositivos com recursos limitados. A EfficientNet-B2, também uma rede convolucional, destaca-se em tarefas mais complexas que demandam maior capacidade de representação, equilibrando acurácia e eficiência sem comprometer significativamente o tempo de execução. Por fim, a ViT-B16, a maior e mais robusta entre as redes analisadas, é uma arquitetura do tipo *transformer* que se destaca pela generalização em tarefas complexas mediante o uso de mecanismos de autoatenção. As diferenças entre os modelos permitem uma análise comparativa não apenas em termos de desempenho, mas também quanto ao custo computacional envolvido em seu treinamento e uso prático.

3. Protocolo Experimental

Para garantir a comparabilidade entre os resultados obtidos, adotou-se um protocolo padronizado de treinamento e avaliação que foi aplicado a todas as redes neurais, conjuntos de dados e métodos de treinamento analisados. Todas as execuções foram feitas na mesma infraestrutura computacional, com os mesmos critérios de parada e configurações de otimização. Essa padronização é fundamental para assegurar que as diferenças observadas no desempenho dos modelos resultem exclusivamente de suas características internas, e não de variações no ambiente ou na condução dos experimentos.

3.1. Técnicas de Treinamento

No treinamento de redes neurais, duas abordagens são amplamente utilizadas: *transfer learning* e *fine-tuning*. Cada uma dessas estratégias apresenta vantagens e desvantagens distintas, estando associadas a diferentes níveis de complexidade computacional, tempo de treinamento e potencial de generalização, dependendo do volume e da natureza dos dados disponíveis.

A técnica de *transfer learning* [11] consiste em congelar todos os pesos da rede pré-treinada, exceto os da última camada (camada de classificação), que é substituída por uma camada adaptada ao número de classes do problema em questão. Essa abordagem é particularmente eficaz quando há alta similaridade entre o domínio de pré-treinamento e o novo domínio. Isso ocorre porque as camadas iniciais do modelo extraem padrões genéricos (bordas, texturas, formas) úteis para a classificação. Por utilizar as representações previamente aprendidas em grandes bases como o ImageNet, o *transfer learning* tende a apresentar boa acurácia com custo computacional reduzido.

Já a abordagem de *fine-tuning* [12] envolve o descongelamento total ou parcial da rede, permitindo que os pesos de todas as camadas sejam ajustados com base no novo conjunto de dados. Embora mais custosa em termos de tempo de treinamento e demanda computacional, essa técnica é indicada para bases de dados maiores ou mais distintas do domínio original do pré-

treinamento. O *fine-tuning* permite que a rede defina suas representações internas, podendo levar a ganhos substanciais de acurácia, especialmente em domínios específicos e complexos.

Em termos comparativos, o *transfer learning* é mais eficiente e rápido, sendo preferido quando se busca uma solução ágil e com recursos computacionais limitados. O *fine-tuning*, por outro lado, apresenta maior capacidade de adaptação e, em muitos casos, resulta em melhores índices de acurácia, desde que haja dados e tempo suficientes para a sua execução.

3.2. Estratégia de Treinamento

Durante o treinamento das redes neurais, o conjunto de dados foi dividido em três subconjuntos: treino, validação e teste. Cada época (*epoch*) representa uma passagem completa pelos dados de treino, que são utilizados para ajustar os parâmetros da rede. Paralelamente, o desempenho do modelo é monitorado em tempo real por meio do subconjunto de validação, permitindo acompanhar sua evolução ao longo das épocas e aplicar técnicas como o *early stopping*. Esta técnica foi empregada neste trabalho como critério de parada antecipada, caso o desempenho do modelo não melhorasse durante cinco épocas consecutivas, com o objetivo de evitar o sobreajuste (*overfitting*) e reduzir o tempo de execução.

Ao final do processo, a avaliação definitiva do modelo é realizada com o conjunto de teste, composto por dados inéditos para a rede, possibilitando uma estimativa realista de sua capacidade de generalização.

A cada época, os pesos da rede são ajustados com base no erro das predições, mensurado por uma função de perda (*loss function*) que avalia a discrepância entre os valores previstos e os valores reais. Quanto menor essa perda, melhor o desempenho do modelo. Para otimizar os ajustes, é utilizado um otimizador que atualiza os parâmetros com base nos gradientes obtidos via retropropagação. Neste trabalho, padronizou-se o uso da função de perda *CrossEntropyLoss*, apropriada para tarefas de classificação multiclasse, juntamente com o otimizador Adam (*Adaptive Moment Estimation*), conhecido por sua robustez e rápida convergência em redes profundas.

Os experimentos foram conduzidos segundo um protocolo estruturado em cinco execuções independentes, cada uma com até 30 épocas. Essa repetição visa lidar com a variabilidade inerente aos algoritmos de aprendizado profundo, permitindo a análise da acurácia média e do desvio padrão dos resultados. Tal abordagem contribui para uma avaliação mais robusta e estatisticamente confiável do desempenho dos modelos testados.

3.3. Métricas de Avaliação

Para cada uma das execuções do experimento, foram coletadas métricas específicas com o objetivo de mensurar o desempenho e a estabilidade dos modelos. As informações quantitativas registradas incluíram:

- **Tempo de treinamento:** Tempo total necessário para treinar o modelo até o critério de parada.
- **Número de épocas:** Total de épocas executadas até a finalização do treinamento, seja por early stopping ou até o limite de 30 épocas.
- **Época com menor perda:** Identificação da época em que se obteve o menor valor da função de perda no conjunto de validação.
- **Tempo de inferência:** Tempo necessário para executar o modelo no conjunto de teste.
- **Acurácia:** Desempenho do modelo em termos de classificação correta no conjunto de teste.

Essas métricas permitiram uma análise abrangente, tanto do desempenho preditivo quanto dos aspectos relacionados ao custo computacional e à estabilidade dos modelos em diferentes condições experimentais.

3.4. Arquitetura de Processamento

Os experimentos foram conduzidos em um servidor Linux equipado com uma GPU **NVIDIA RTX A5000**, com **24 GB** de memória dedicada e suporte à arquitetura CUDA na versão 12.5. Todo fluxo de treinamento e inferência foi implementado utilizando o *framework* PyTorch do Python, amplamente adotado na comunidade científica por sua flexibilidade, integração com CUDA e suporte a operações vetoriais otimizadas.

4. Resultados Obtidos

A acurácia, métrica adotada para mensurar o desempenho dos modelos, foi registrada em cada execução dos experimentos com o intuito de avaliar tanto o melhor resultado alcançado quanto a estabilidade entre as execuções. A Tabela 2 apresenta as médias das acurácias obtidas, acompanhadas dos respectivos desvios padrão. De modo geral, os baixos valores de desvio padrão observados indicam que os modelos apresentaram comportamentos consistentes ao longo das execuções avaliadas.

Observa-se que o CIFAR-10 apresentou, em média, o maior desvio padrão, seguido pelo FashionMNIST. Essa diferença pode ser atribuída à maior complexidade do CIFAR-10, cujas imagens coloridas englobam uma diversidade significativa de objetos, ângulos e contextos visuais, enquanto o FashionMNIST é composto por figuras em tons de cinza e com menor variabilidade estrutural. Em contraste, o PKLot apresentou o menor desvio padrão entre os conjuntos avaliados, sugerindo maior estabilidade nos resultados — possivelmente devido à natureza binária do dataset.

Outro aspecto notável é que, na maioria dos casos, o método de treinamento *transfer learning* apresentou maior estabilidade nas acurácias em comparação ao *fine-tuning*, coerente com a natureza dos dois métodos.

Tabela 2: Média da acurácia das execuções dos experimentos e seus respectivos desvios padrão (em percentual). A média geral por dataset também é apresentada na linha de cabeçalho de cada bloco..

Neural Network	Fine-Tuning	Transfer Learning
FashionMNIST - 90,20 (0,292)		
MobileNetV3 Small	93,32 (0,342)	89,29 (0,230)
EfficientNet-B2	**93,88** (0,218)	85,15 (0,203)
ViT-B16	88,97 (0,600)	90,56 (0,179)
CIFAR-10 - 85,28 (0,526)		
MobileNetV3 Small	90,76 (0,625)	85,51 (0,303)
EfficientNet-B2	93,70 (0,412)	82,10 (0,240)
ViT-B16	65,43 (1,478)	**95,36** (0,097)
PKLot - 99,73 (0,025)		
MobileNetV3 Small	**99,89** (0,018)	99,77 (0,036)
EfficientNet-B2	99,88 (0,033)	99,18 (0,031)
ViT-B16	99,87 (0,015)	99,80 (0,017)

Compreendida a estabilidade dos modelos, direciona-se a análise aos desempenhos obtidos ao longo dos experimentos. Ao observar os resultados das acurácias médias apresentadas, nota-se que não houve uma única arquitetura de rede neural que se destacasse de forma absoluta em todos os conjuntos de dados. Cada dataset favoreceu uma combinação distinta de arquitetura e método, indicando que o desempenho ótimo depende fortemente das características do conjunto de dados em relação à capacidade de representação de cada rede.

As arquiteturas MobileNetV3 Small e EfficientNet-B2 apresentaram um padrão consistente: em todos os casos, o *fine-tuning* superou o *transfer learning*, com diferenças expressivas de acurácia. Esses resultados sugerem que foi possível ajustar integralmente essas redes, permitindo que elas se adaptassem de maneira eficaz aos dados específicos de cada tarefa. A estrutura mais compacta dessas redes, quando comparada a arquiteturas mais profundas, pode ter favorecido esse processo de ajuste completo, sem comprometer a generalização.

Por outro lado, o comportamento da ViT-B16 foi distinto. Em dois dos três conjuntos de dados (FashionMNIST e PKLot), o *transfer learning* superou o *fine-tuning*. Esse resultado pode estar relacionado à maior complexidade e profundidade da arquitetura ViT, que demanda um volume maior de dados ou mais tempo de treinamento para ser plenamente ajustada. É importante ressaltar que o número de parâmetros da ViT-B16 ultrapassa significativamente o número de amostras disponíveis para o treinamento, o que pode ter limitado o seu potencial de generalização no ajuste completo.

Esses resultados reforçam a importância de considerar as características arquiteturais dos modelos na definição da estratégia de ajuste. Enquanto redes menores podem se beneficiar do *fine-tuning* completo, redes mais profundas e sensíveis, como *transformers*, podem demandar ajustes mais específicos nos critérios de parada e nos hiperparâmetros de treinamento para atingirem seu desempenho ideal.

Além do desempenho em termos de acurácia, outro aspecto fundamental para a avaliação dos modelos é o custo computacional associado ao seu uso. A Figura 4 apresenta um gráfico com os tempos médios de inferência de cada imagem, obtidos em cada ciclo de execução dos experimentos.

Observa-se uma notável similaridade no tempo de inferência entre os ciclos de cada rede neural, independentemente do conjunto de dados ou do método de treinamento utilizado. Isso acontece pois, durante a inferência, todas as imagens são previamente norma-

lizadas e padronizadas, e a estrutura arquitetural de cada rede permanece inalterada após o treinamento — ou seja, o número de camadas e de parâmetros não varia entre as execuções.

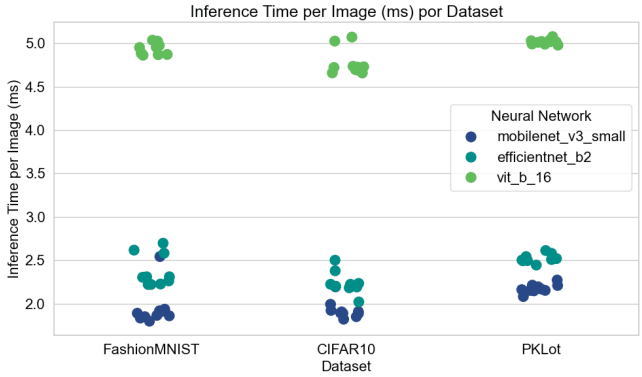

Figura 4: Tempo de inferência por imagem em cada experimento.

Pequenas flutuações são observadas nos tempos e devem-se principalmente a fatores de ambiente, como variações no processamento do hardware e diferenças marginais nos pesos ajustados ao longo dos ciclos. No entanto, para cada rede, o tempo de inferência apresentou valores altamente consistentes, reforçando a previsibilidade do custo computacional.

Complementando essa análise visual, a Tabela 3 apresenta os valores médios de tempo de inferência por imagem. Observa-se que a MobileNetV3 Small apresentou o menor tempo médio, seguida pela EfficientNet-B2 e pela ViT-B16, que apresentou o maior tempo médio. Essa ordenação está em total consonância com a expectativa teórica, já que reflete diretamente o grau de complexidade e o volume de parâmetros de cada arquitetura.

Tabela 3: Tempo médio de inferência por imagem para cada rede neural.

Neural Network	Tempo de Inferência por Imagem
MobileNetV3 Small	2.014 ms
EfficientNet-B2	2.378 ms
ViT-B16	4.905 ms

Vale relembrar que todos os experimentos de inferência foram realizados em uma GPU de alto desempenho, com uma arquitetura otimizada para operações massivamente paralelas. Essa característica permite que modelos maiores, como o ViT-B16, se beneficiem fortemente do paralelismo computacional, reduzindo o tempo por inferência mesmo diante de um número significativamente maior de parâmetros. Isso ajuda a explicar por que, neste cenário, a ViT-B16 apresentou um tempo de inferência apenas cerca de duas vezes superior ao das demais redes. No entanto, em ambientes com hardware mais modesto, essa diferença tenderia a ser substancialmente maior, o que evidencia a importância de considerar a infraestrutura computacional disponível na escolha da arquitetura a ser utilizada em aplicações práticas.

Por fim, avaliou-se, para cada experimento, o tempo total de treinamento, somando-se o tempo de treinamento e a quantidade de épocas de cada uma das cinco execuções.

Com o auxílio da Figura 5, observa-se que o tempo de treinamento seguiu uma ordem crescente entre as arquiteturas: MobileNetV3 Small, EfficientNet-B2 e ViT-B16. Essa ordenação reflete diretamente a complexidade crescente das redes e a quantidade progressivamente maior de parâmetros em cada modelo. Essa relação manteve-se consistente em todos os datasets avaliados.

Outro aspecto importante foi a comparação entre os métodos de treinamento. Em seis dos nove experimentos, o *transfer learning* apresentou menor tempo total de treinamento em relação ao *fine-tuning*. Entretanto, em três casos específicos, ocorreu o contrário. Esse resultado, à primeira vista contraintuitivo, decorre do fato de que o número total de épocas executadas variou de forma significativa em função do critério de *early stopping*. Nessas situações, o *transfer learning* precisou de mais iterações até a convergência do modelo, compensando o tempo adicional que o *fine-tuning* demanda em cada época individual.

Por fim, é importante destacar a ampla variação do tempo total de treinamento entre os experimentos. O menor tempo foi registrado na configuração MobileNetV3 Small – CIFAR10 – *transfer learning*, que completou o processo em aproximadamente uma hora. Em contraste, o maior tempo foi observado na configuração ViT-B16 – PKLot – *fine-tuning*, que demandou mais de 70 horas de processamento até a finalização. Essa disparidade evidencia como a escolha da arquitetura e da estratégia de ajuste impacta de maneira substancial a viabilidade operacional do uso dos modelos em ambientes com restrições de recursos e prazos.

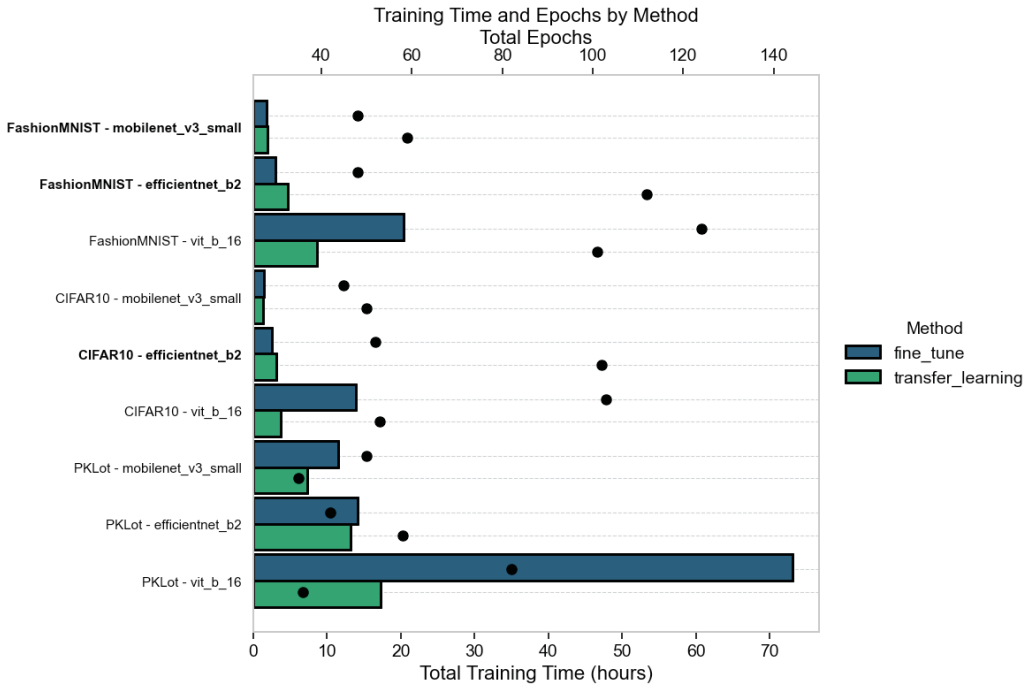

Figura 5: Representação gráfica do tempo de treinamento com barras (escala inferior) e da quantidade de épocas com pontos (escala superior)

5. Conclusão

Os experimentos conduzidos neste estudo evidenciam que, considerando os conjuntos de dados e as arquiteturas avaliadas, não foi possível identificar uma única combinação de rede neural e método de treinamento que se destacasse de forma absoluta em todos os cenários. Cada dataset favoreceu abordagens distintas, tanto em termos de desempenho preditivo quanto de viabilidade computacional. Esses resultados ilustram que, mesmo dentro de um domínio restrito, diferentes arquiteturas e estratégias de ajuste podem apresentar desempenhos contrastantes.

O tempo de inferência, por sua vez, pode ser uma variável determinante na escolha de um modelo em aplicações práticas. Embora este estudo tenha evidenciado que as arquiteturas apresentam tempos de predição significativamente distintos , destaca-se que, em cenários reais de classificação de imagens, a latência frequentemente assume importância equivalente ou até superior à própria acurácia quando o desempenho preditivo entre diferentes modelos é comparável. Em aplicações que exigem processamento em tempo real ou elevada taxa de requisições simultâneas, essa limitação pode inviabilizar arquiteturas mais complexas, mesmo quando apresentam ganhos de acurácia moderados. Por outro lado, não se deve negligenciar que, abaixo de determinados patamares de desempenho, a precisão do modelo torna-se indispensável, independentemente de sua velocidade de execução.

O tempo de treinamento também se mostrou altamente variável, com diferenças substanciais entre as arquiteturas das redes neurais, *datasets* e métodos de treinamento. Ao todo, foram necessários cerca de nove dias contínuos de processamento para a execução completa dos experimentos.

Em síntese, os resultados obtidos neste estudo demonstraram que, dentro do escopo dos experimentos conduzidos, não existe uma solução universalmente ótima que concilie, ao mesmo tempo, máxima acurácia, menor tempo de treinamento e eficiência de inferência. A escolha do modelo e da estratégia de ajuste deve ser orientada por uma avaliação equilibrada de múltiplos fatores, incluindo o domínio do problema, as restrições de infraestrutura, os requisitos de atualização periódica dos pesos e as demandas de latência operacional. Cada problema de classificação de imagens possui especificidades que requerem experimentação prévia e análise comparativa criteriosa para que se obtenha uma solução técnica adequada e sustentável.

Referências

[1] Y. LeCun, Y. Bengio, and G. Hinton, *Deep learning*, Nature, vol. 521, no. 7553, pp. 436–444, 2015.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, *ImageNet classification with deep convolutional neural networks*, Communications of the ACM, vol. 60, no. 6, pp. 84–90, 2017.

[3] E. Strubell, A. Ganesh, and A. McCallum, *Energy and Policy Considerations for Deep Learning in NLP*, Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 3645–3650, 2019.

[4] R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni, *Green AI*, Communications of the ACM, vol. 63, no. 12, pp. 54–63, 2020.

[5] H. Xiao, K. Rasul, and R. Vollgraf, *Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms*, arXiv preprint arXiv:1708.07747, 2017.

[6] A. Krizhevsky, *Learning Multiple Layers of Features from Tiny Images*, Technical Report, University of Toronto, 2009.

[7] P. H. de Almeida, L. S. Oliveira, A. L. Koerich, *PKLot – A robust dataset for parking lot classification*, Expert Systems with Applications, vol. 42, no. 11, pp. 4937–4949, 2015.

[8] A. Howard, M. Sandler, G. Chu, L. Chen, B. Chen, M. Tan, G. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le, and H. Adam, *Searching for MobileNetV3*, Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 1314–1324, 2019.

[9] M. Tan and Q. V. Le, *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*, Proceedings of the 36th International Conference on Machine Learning (ICML), vol. 97, pp. 6105–6114, 2019.

[10] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*, International Conference on Learning Representations (ICLR), 2021. arXiv:2010.11929.

[11] S. J. Pan and Q. Yang, *A Survey on Transfer Learning*, IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 10, pp. 1345–1359, 2010.

[12] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, *How Transferable Are Features in Deep Neural Networks?*, Advances in Neural Information Processing Systems (NeurIPS), vol. 27, pp. 3320–3328, 2014.

[13] 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.