Projeto 5 - Redes Convolucionais

Guilherme Pereira Campos RA:163787
Lucas Oliveira Nery de Araújo RA:158882
Universidade Federal de São Paulo

I. Introdução

Este projeto tem como objetivo implementar e avaliar diferentes arquiteturas de redes *CNN* para a classificação de imagens do dataset *MNIST*. Serão testadas cinco topologias clássicas: **LeNet**, **AlexNet**, **VGG**, **GoogLeNet** e **ResNet**, analisando o impacto da profundidade, número de filtros e hiperparâmetros no desempenho.

Os dois modelos com melhor acurácia serão analisados em detalhes, incluindo a geração da matriz de confusão para avaliar o desempenho por classe. Além disso, será feita uma comparação entre suas arquiteturas e eficiência na classificação.

Por fim, uma rede *MLP*, previamente treinada no mesmo conjunto de dados, será utilizada como referência para comparar a acurácia e o número de parâmetros em relação à melhor *CNN*, destacando as vantagens e limitações de cada abordagem.

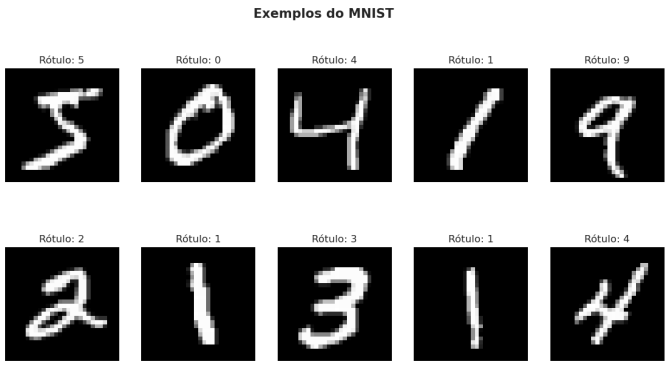
II. Dataset

O dataset utilizado é o *MNIST (Modified National Institute of Standards and Technology)*, ele contém 70.000 imagens em escala de cinza, com tamanho de 28x28 *pixels*, representando dígitos manuscritos de 0 a 9. O *dataset* é dividido em 60.000 imagens de treinamento e 10.000 de teste.

A. Preparação dos dados

A seguinte preparação foi feita:

- **Normalização das Imagens:**
  - Os pixels originalmente variam de **0 a 255**.
  - Para melhorar a estabilidade do modelo, os valores são convertidos para **o intervalo [0,1]**, usando divisão por 255.
- **Separação dos Conjuntos:**
  - Para avaliar o desempenho do modelo antes do teste, **separamos 5.000 imagens do treino para validação**.
  - Distribuição final dos dados:
    - **Treino:** 55.000 imagens
    - **Validação:** 5.000 imagens
    - **Teste:** 10.000 imagens
- **Visualização dos Dados:**
  - São exibidas **10 imagens do conjunto de treino** com seus respectivos rótulos.
  - Isso permite verificar a qualidade e distribuição dos dados antes do treinamento.



III. Redes convolucionais

A. Definição

As redes convolucionais (*CNN*) são uma classe de redes neurais especialmente eficazes para o processamento de dados estruturados, como imagens, séries temporais e outros. Elas utilizam operações de convolução para extrair características locais dos dados, reduzindo o número de parâmetros e melhorando a generalização.

B. Campos Receptivos Locais

- O cérebro não analisa a imagem de forma global; ao invés disso, células específicas se concentram em regiões específicas, formando os chamados *campos receptivos locais*.
- Essa abordagem é a base das **Redes Convolucionais (CNN)**.

C. Comparação com Redes MLP

- Ao contrário das redes *MLP*, as *CNNs* utilizam campos receptivos locais, e não globais.
- Há replicação dos neurônios em um mesmo filtro.
- Isso resulta em uma significativa redução do número de parâmetros livres.

D. Convolução

A operação de convolução pode ser definida como:

$$f_{og}(x) = \sum_{k=-M}^{M} f(x-k)\, g(k)$$

- Em redes convolucionais, a convolução pode ser aplicada a múltiplos canais.

E. Camada de Pooling

- A camada de *Pooling* reduz o mapa de características gerado pela camada convolucional.
- Diminui o custo computacional do modelo.
- Introduz invariância à translação, melhorando a robustez do modelo.

F. Stride e Padding

- **Stride**: deslocamento da janela durante a aplicação da convolução (por exemplo, *stride* = 1).
- **Padding**: preenchimento das bordas da imagem para controlar o tamanho do mapa de características resultante.
  - Pode ser definido como 0 (sem preenchimento), por replicação, com um valor constante ou de forma circular.

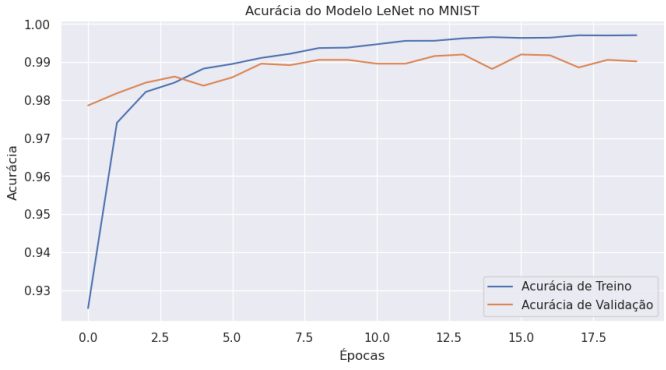
G. Aplicações das CNN

- **Visão Computacional**:
  - Classificação de imagens.
  - Reconhecimento e localização de objetos.
  - Segmentação, entre outras tarefas.
- **Séries Temporais**:
  - Convolução 1D para processamento de sequências numéricas.
- **Processamento de Textos e Áudio**:
  - Aplicações em processamento de linguagem natural e análise de sinais de áudio.

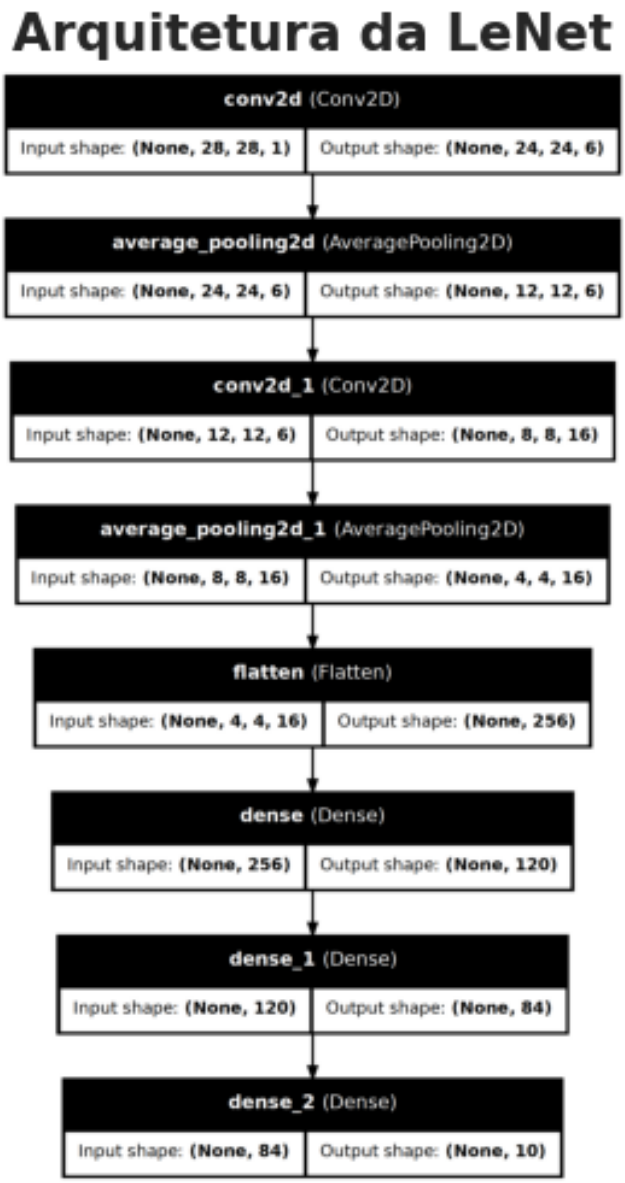
IV. Topologias

A. LeNet

A LeNet é uma rede convolucional clássica projetada para reconhecimento de dígitos no MNIST. Ela utiliza duas camadas convolucionais com ativação *ReLU* e *pooling* médio para extrair características, seguidas por camadas densas que realizam a classificação. Essa arquitetura simples captura padrões locais essenciais de imagens 28x28 em escala de cinza. A camada final usa *softmax* para gerar probabilidades para 10 classes. O modelo demonstra eficiência e baixa complexidade para tarefas de classificação.



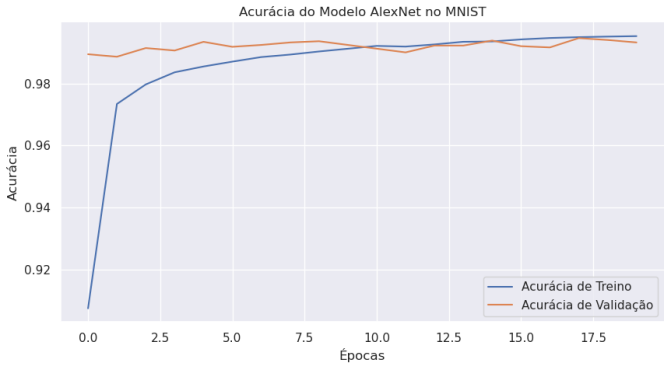
Apesar de sua simplicidade, a acurácia da LeNet é boa, demonstrando que a arquitetura é eficaz para tarefas de classificação de imagens, como o reconhecimento de dígitos no MNIST. A combinação de camadas convolucionais e pooling médio permite capturar padrões locais de forma eficiente, enquanto a camada densa final realiza uma classificação robusta.



B. AlexNet

A *AlexNet* é uma rede convolucional profunda com 8 camadas treináveis, sendo 5 convolucionais e 3 totalmente conectadas. Possui cerca de 60 milhões de parâmetros e 650 mil neurônios, utilizando *ReLU* como ativação e *dropout* para evitar overfitting. A arquitetura inclui camadas de *pooling* para redução da dimensionalidade e normalização local para estabilizar o treinamento.

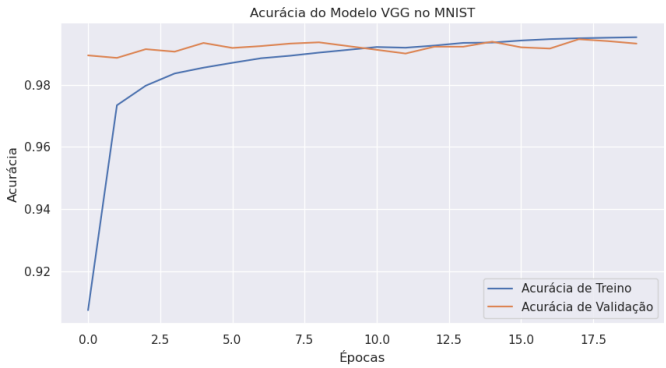
A *AlexNet* tem uma acurácia muito boa, sendo uma das redes mais eficazes para tarefas de classificação. Sua complexidade



é mediana, com um número considerável de parâmetros e camadas treináveis, o que a torna poderosa, mas sem ser excessivamente complexa. Apesar de ser mais complexa do que redes mais simples, como a *LeNet*, ela entrega um ótimo desempenho em diversas tarefas de visão computacional.

C. VGG (Visual Geometry Group)

A *VGG* (Visual Geometry Group) é uma rede convolucional profunda caracterizada pelo uso de camadas convolucionais empilhadas com filtros pequenos (3×3) e camadas de pooling (2×2) para redução da dimensionalidade. A arquitetura segue um padrão simples, aumentando progressivamente a profundidade da rede enquanto reduz a dimensão espacial das features. Utiliza *ReLU* como ativação e camadas totalmente conectadas no final para classificação.

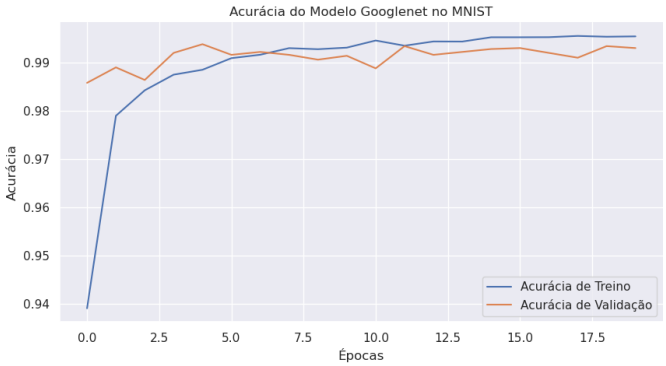


A *VGG* é uma rede mais complexa, com uma arquitetura mais profunda e com maior número de parâmetros em comparação com a *LeNet* e a *AlexNet*. No entanto, sua acurácia foi extremamente boa, demonstrando sua eficácia em tarefas de classificação.

D. GoogleNet

A *GoogLeNet* é uma arquitetura de rede neural profunda que utiliza módulos chamados "Inception". Cada módulo combina várias operações convolucionais com diferentes tamanhos de kernel (1x1, 3x3, 5x5), além de pooling, permitindo que a rede capture informações em diferentes escalas. A *GoogLeNet* introduziu o uso de convoluções 1x1 para reduzir a dimensionalidade e melhorar a eficiência computacional, tornando-se mais profunda e leve. Ela também utiliza um conceito de "auxiliary classifiers" para ajudar no treinamento.

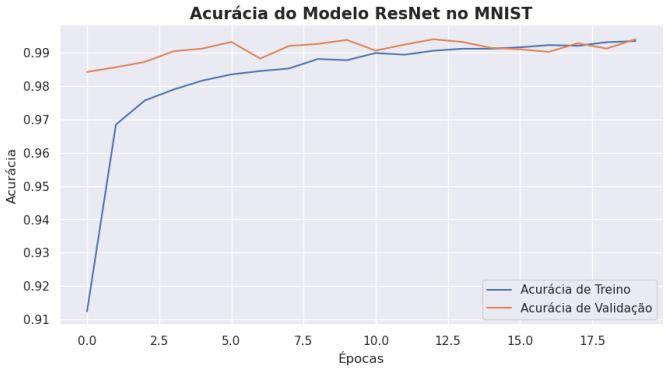
A *GoogLeNet* teve um desempenho bom, mas não foi ótima, apesar de sua complexidade. A arquitetura profunda com os módulos Inception é eficiente, permitindo que a rede capture informações em diferentes escalas. Porém, mesmo com essa flexibilidade, seu desempenho não foi o melhor quando comparado com outras redes, seja mais simples ou até mais complexas.



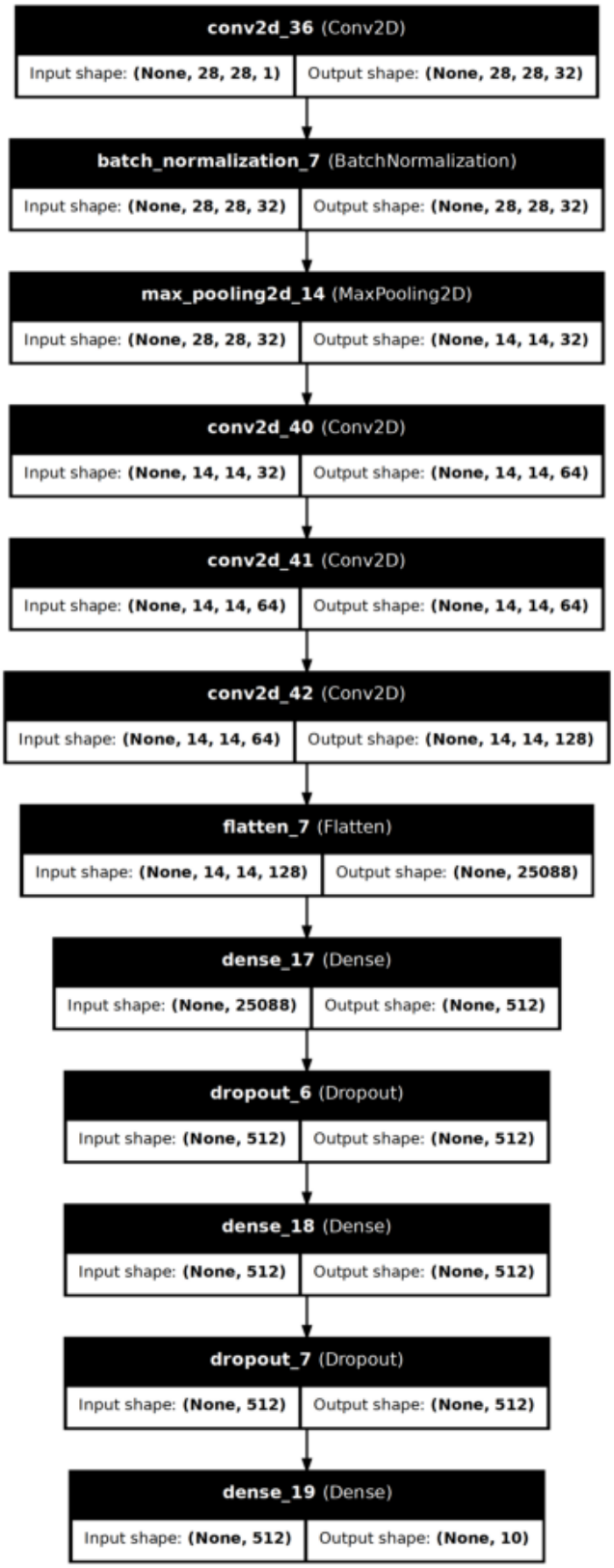
E. ResNet (Residual Neural Network)

A *ResNet* é uma arquitetura de rede profunda que utiliza conexões residuais, permitindo que as camadas saltem uma ou mais etapas, facilitando o fluxo de gradientes e mitigando o problema de degradação. Essas conexões ajudam a manter a precisão mesmo em redes muito profundas. A *ResNet* é composta por blocos residuais repetidos que adicionam a entrada à saída da camada.

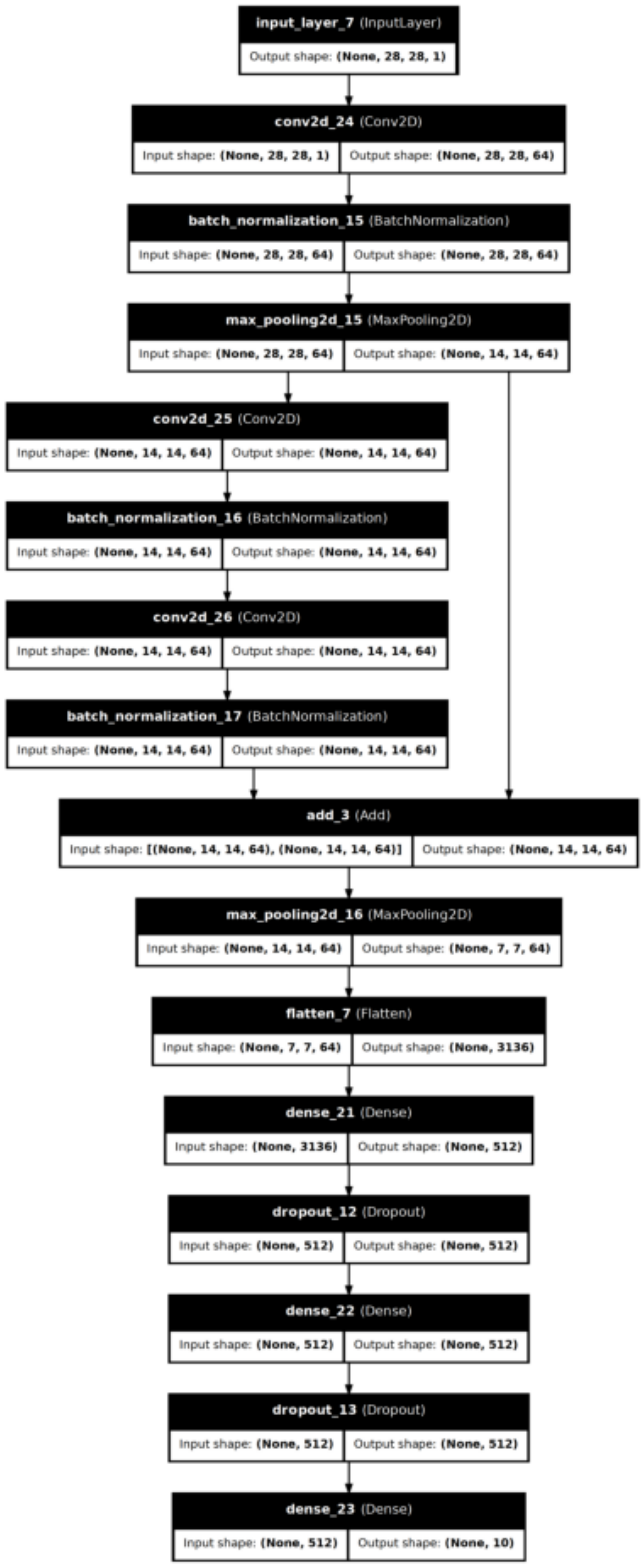
A ResNet é uma rede profunda que usa conexões residuais, permitindo que as camadas "saltem" algumas etapas. Isso ajuda no fluxo de gradientes e evita o problema de degradação em redes muito profundas. Embora seja bastante complexa, a ResNet mantém boa precisão, mas seu desempenho não é tão bom quanto o de modelos como AlexNet e VGG.

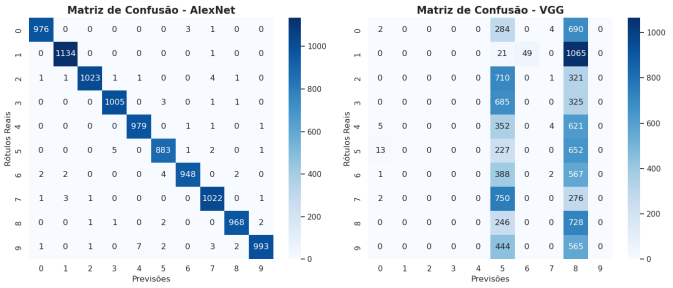


Arquitetura do Googlenet



Arquitetura do ResNet





V. Melhores modelos

Os dois melhores modelos foram ***AlexNet*** (0.9937) e ***VGG*** (0.9908).

As principais diferenças entre a AlexNet e a VGG estão na profundidade da rede e na forma como as convoluções são aplicadas. A AlexNet possui menos camadas convolucionais e utiliza filtros maiores, o que permite um processamento mais rápido durante o treinamento. Já a VGG adota uma abordagem mais profunda, com múltiplas camadas convolucionais seguidas e filtros menores (3x3), o que melhora a extração de características, mas aumenta o tempo de treino. Enquanto a AlexNet é mais rápida para treinar, a VGG pode oferecer melhor desempenho em tarefas complexas devido à sua maior capacidade de aprendizado.

VI. Comparação entre MLP implementada em projeto anteior

O modelo *mlp_antiga*, baseado em LSTM, foi treinado em um trabalho anterior. Sua arquitetura inclui três camadas LSTM com 64, 32 e 16 neurônios, seguidas por Dropout (20%) e uma camada Dense com softmax. Agora, esse modelo será comparado com as melhores CNNs.

Apesar de apresentar uma acurácia razoável, seu desempenho foi pior que as arquiteturas CNN, como **AlexNet** e **VGG**. As CNNs extraem características espaciais por meio de camadas convolucionais, capturando detalhes e invariâncias que as LSTMs, voltadas para dados sequenciais, não conseguem explorar da mesma forma. Essa comparação ressalta que, para tarefas de classificação de imagens, as CNNs podem oferecer uma abordagem melhor.

VII. Conclusão

Em conclusão, este projeto explorou diferentes arquiteturas de redes convolucionais (*CNN*) para classificar imagens do dataset *MNIST*. Testamos cinco modelos: **LeNet**, **AlexNet**, **VGG**, **GoogLeNet** e **ResNet**, analisando como a profundidade, o número de filtros e os hiperparâmetros afetam o desempenho. Os dois melhores modelos foram **AlexNet** (*0.9937*) e **VGG** (*0.9908*), que apresentaram acurácia superior aos outros. Eles foram avaliados com mais detalhes, usando a matriz de confusão para examinar o desempenho por classe. A comparação entre as arquiteturas e a eficiência de cada uma revelou pontos fortes e fracos de cada abordagem. Por fim, comparamos esses modelos com uma rede *MLP* previamente treinada, destacando as diferenças na acurácia e no número de parâmetros.