



CN's



Gabriel Ricetto, Joao Vitor Pastori e Victor Querino Martins

Definição

O problema de classificação de imagens escolhido é a identificação e categorização automática de diferentes tipos de veículos - especificamente carros, caminhões e vans - a partir de imagens. Este é um problema clássico de classificação multiclasse, dada a sua alta capacidade de extrair características complexas e hierárquicas de dados visuais.



Aplicação



* No Trânsito

Otimizar o fluxo de veículos, fiscalizar zonas de restrição e ajudar no planejamento de manutenção das ruas, identificando o desgaste causado por veículos pesados.

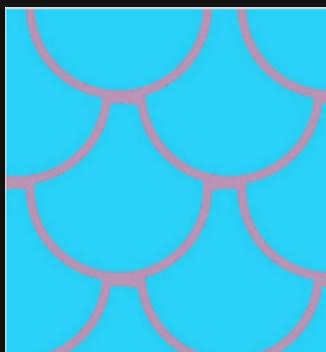
* Na Segurança

Acelerar a resposta a emergências, informando às equipes de resgate exatamente quais tipos de veículos estão envolvidos em um acidente.



Dataset

Para este projeto, utilizamos um dataset do site kaggle . Onde se encontra várias imagens de modelos diversos de carros caminhões e vans, ensinando-o a distinguir as características visuais que definem carros, vans e caminhões.



Vehicle Images Dataset
Kaggle is the world's largest data science community with powerful tools and resources to help you achieve your data science goals.
[kaggle.com](https://www.kaggle.com)





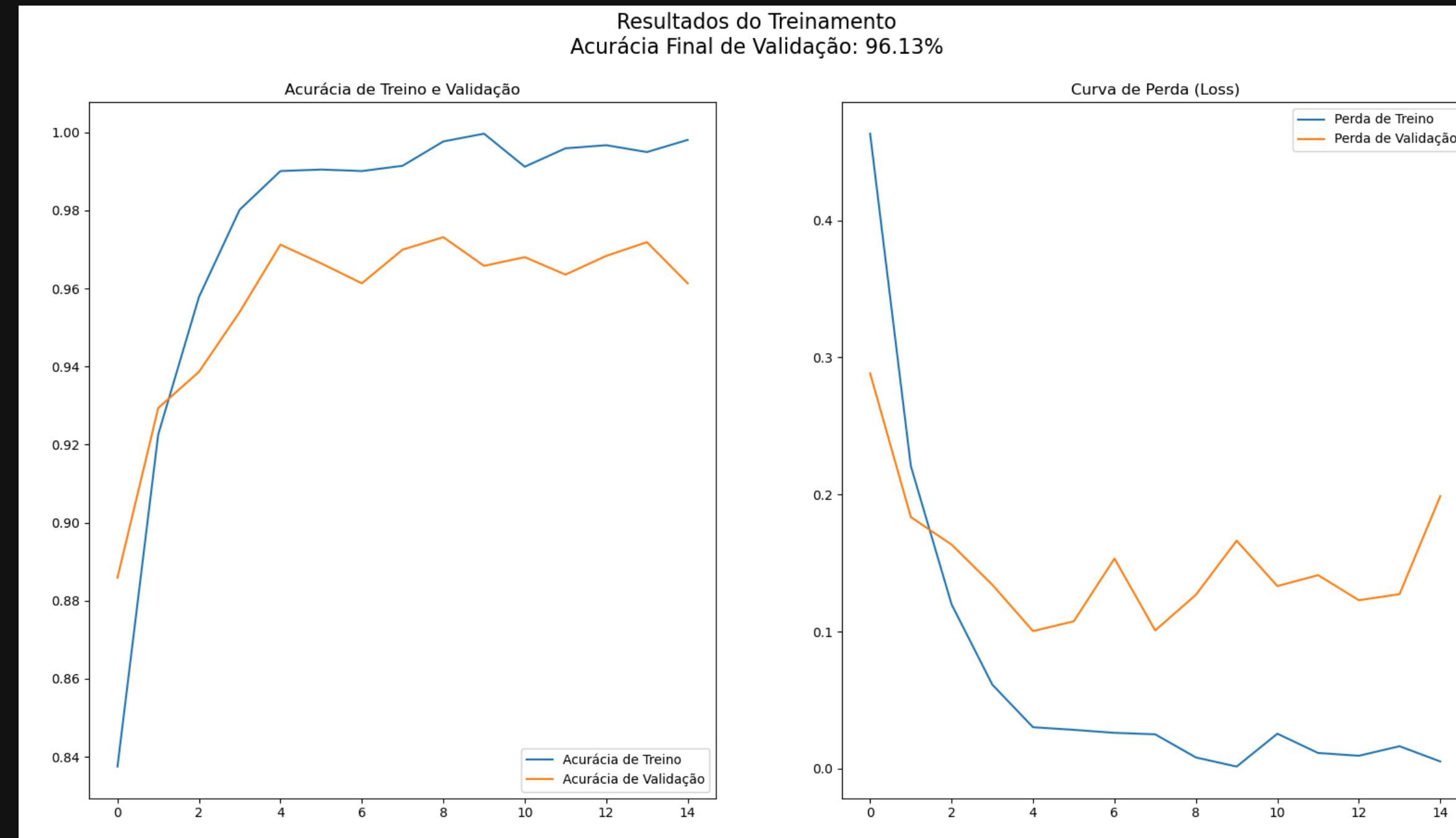
PARA EXECUÇÃO:

- Diminuição do Tamanho da Imagem:
`IMG_SIZE` foi alterado de **(128, 128)** para **(96, 96)**.
- Redução do Número de Épocas:
`FIXED_EPOCHS` foi alterado de **10** para **5**.

U

LENET

Acurácia



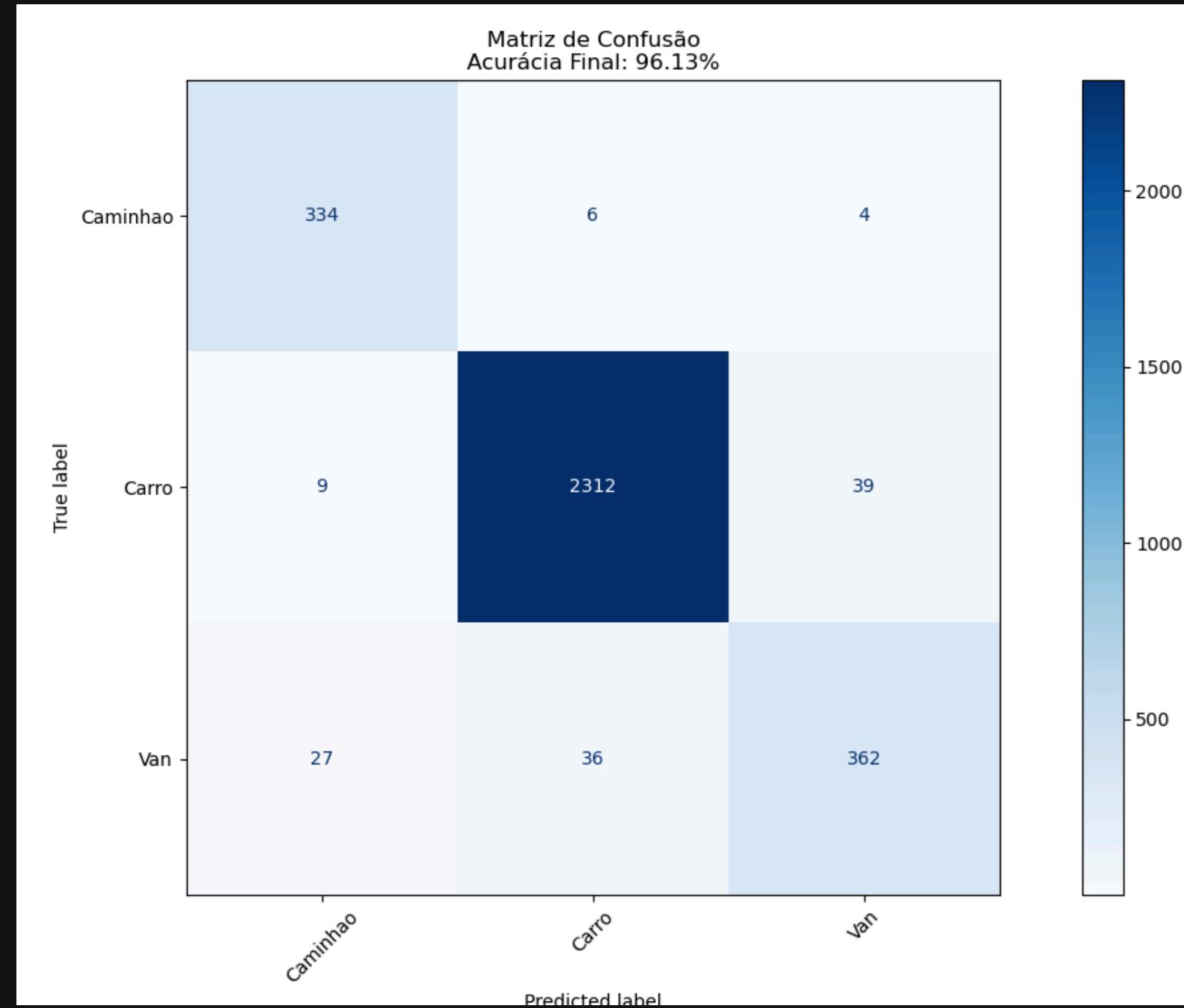
Acurácia: 96,13% Loss: 0.1989





LENET

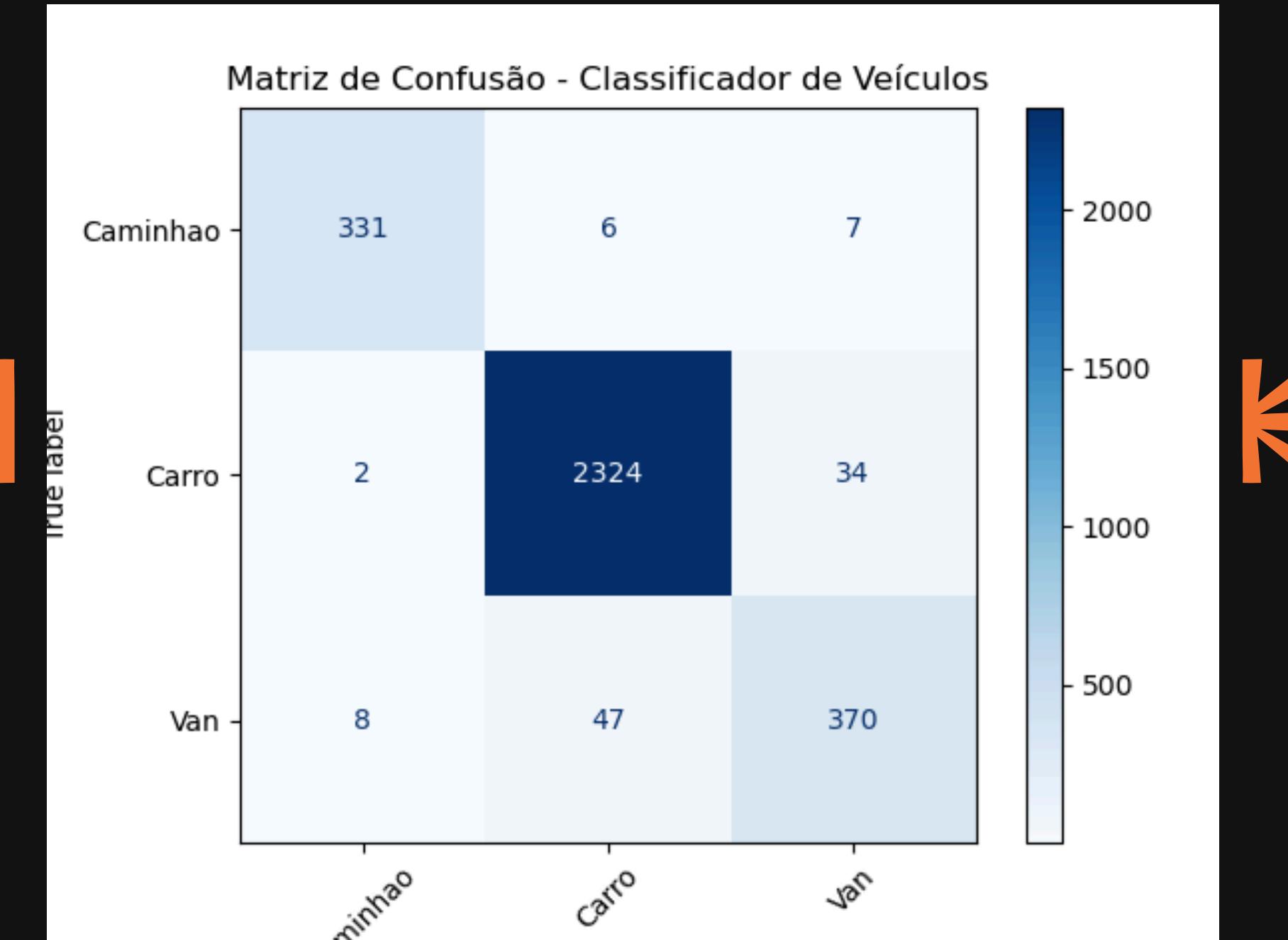
Matrix De Confusão



Acurácia: 96,13% Loss: 0.1989

ALEXNET

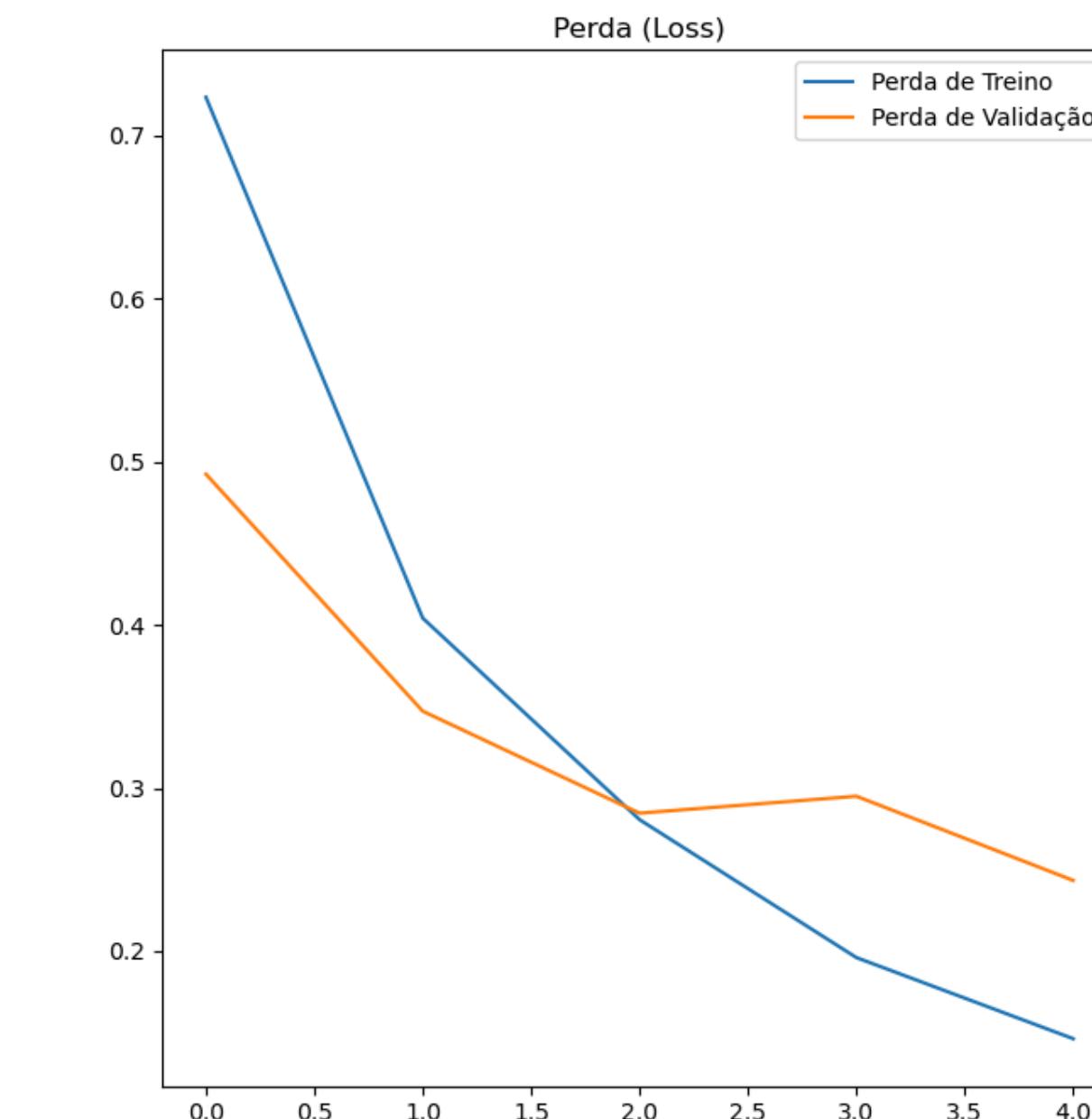
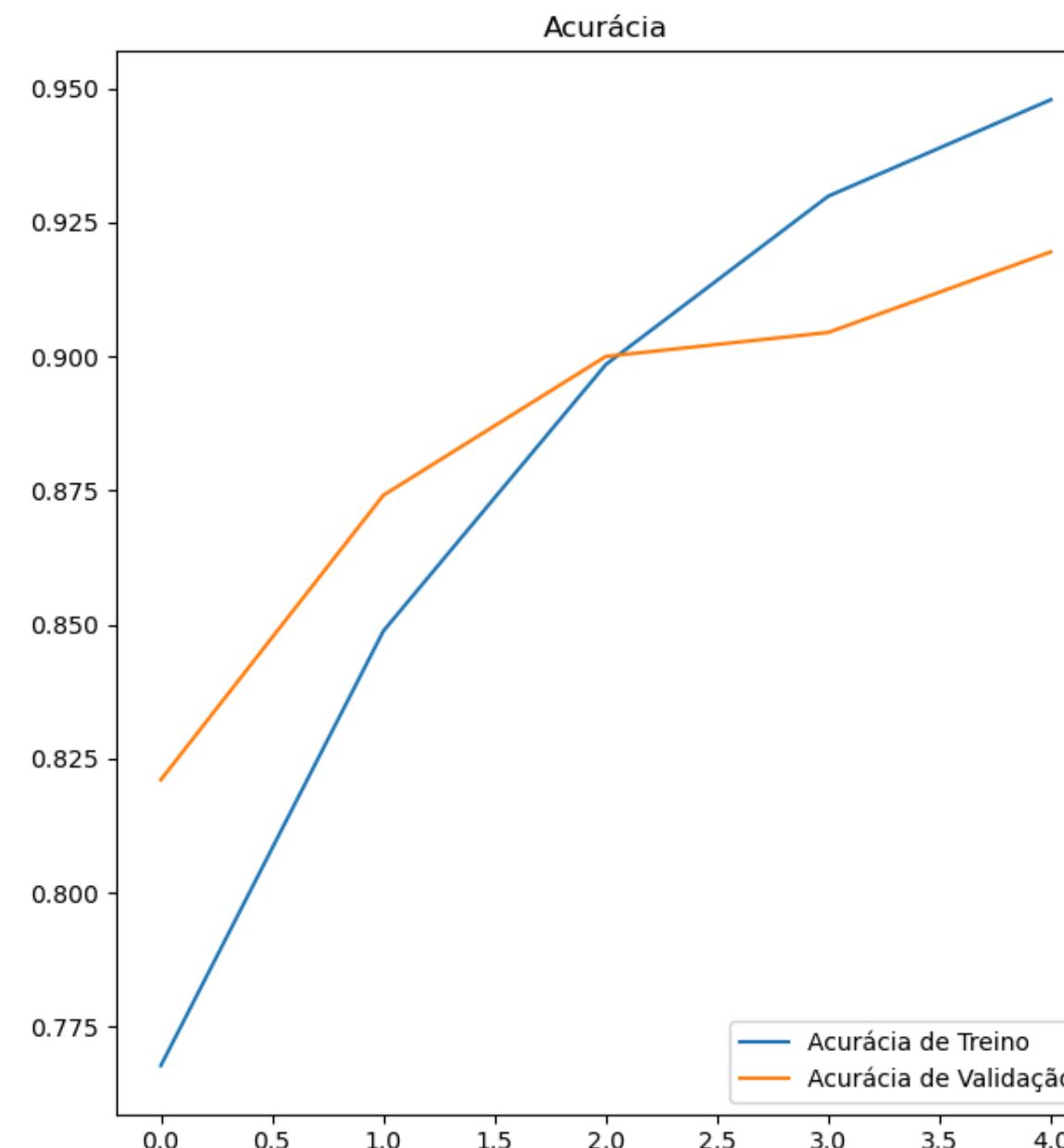
Matrix De Confusão



Acurácia: 95,83% Loss: 0.1179

Auto-Tuning

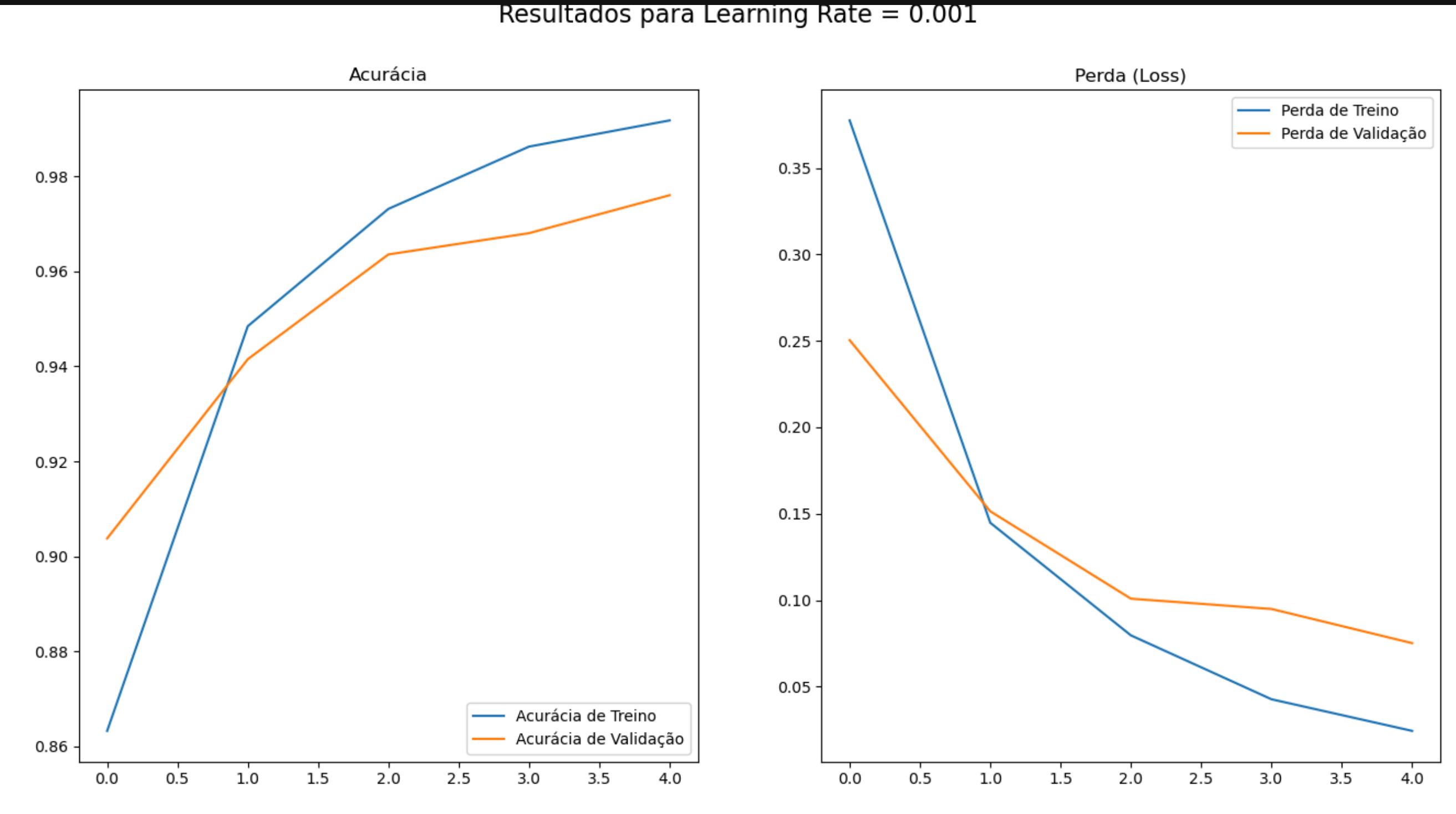
Resultados para Learning Rate = 0.01



Learning Rate 0.01

Auto-Tuning

Resultados para Learning Rate = 0.001

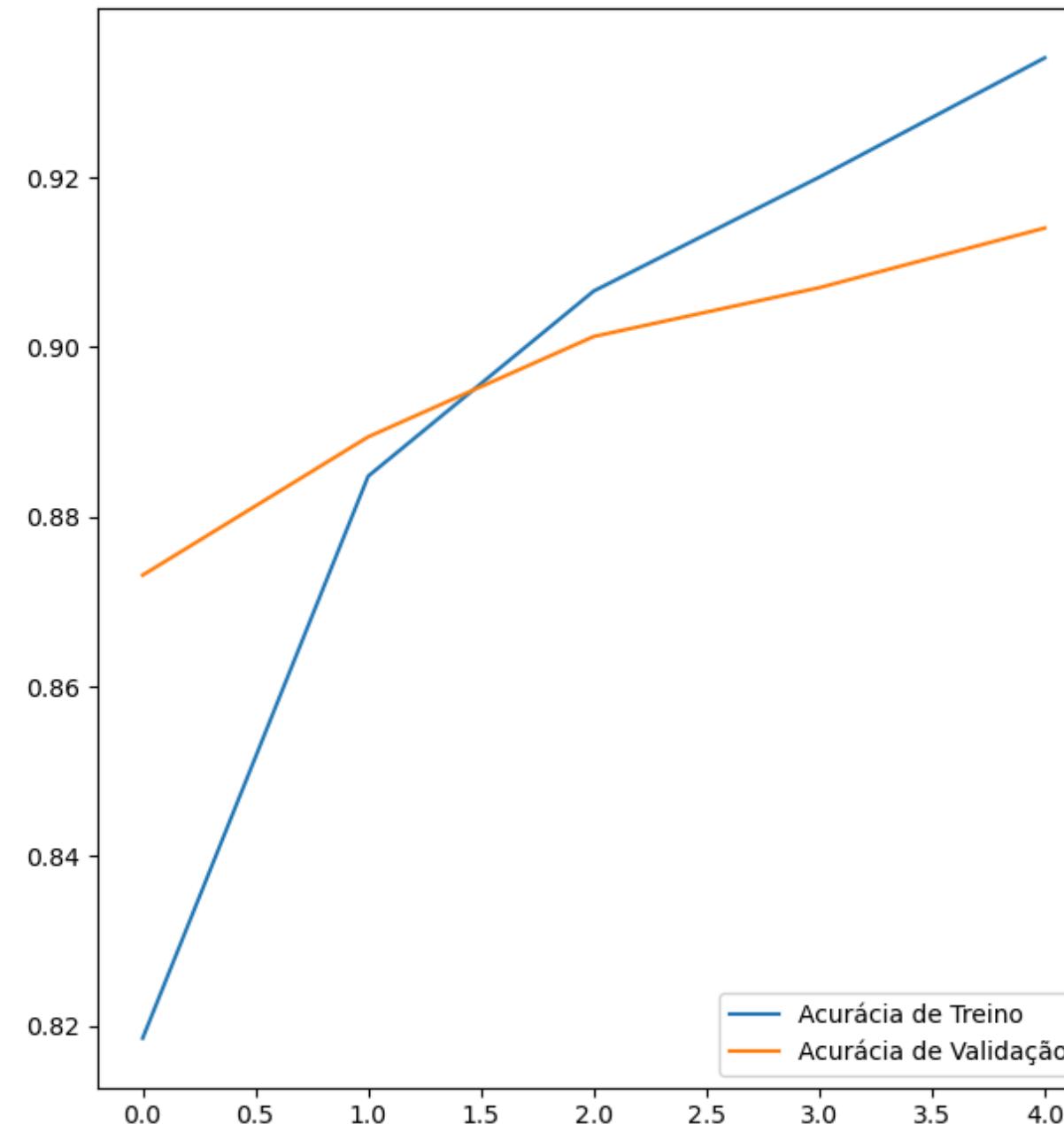


Learning Rate 0.001

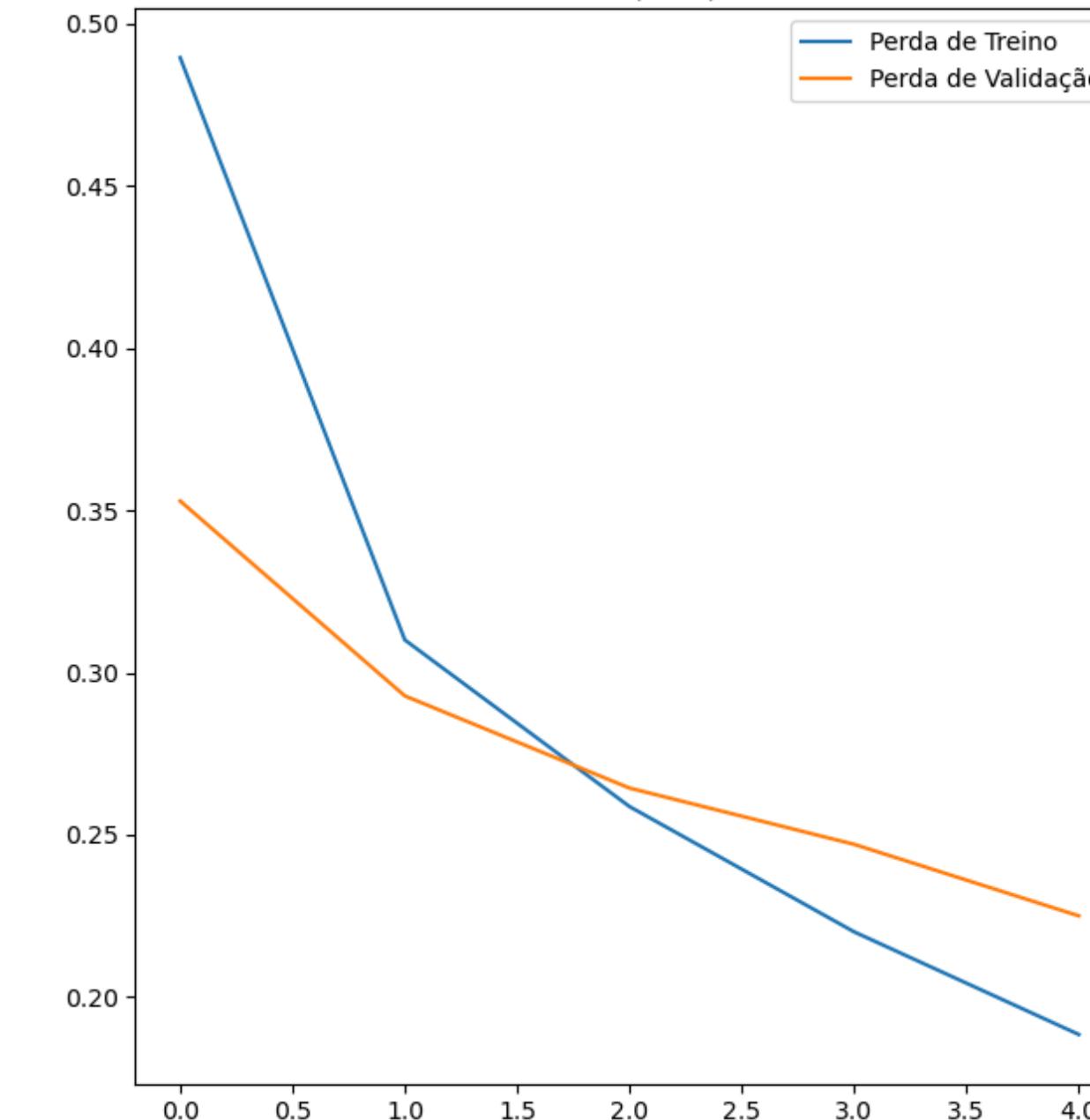
Auto-Tuning

Resultados para Learning Rate = 0.0001

Acurácia



Perda (Loss)



Learning Rate 0.0001



LEARNING RATE

0.01

- Resultado **Ruim**. O modelo aprendeu rápido demais e começou a "decorar" os dados de treino.
- O sinal claro disso é que a curva de **Loss** começou a subir, indicando que o modelo estava piorando seu desempenho em imagens novas .

0.001

- **Ótimo** Resultado. Este é o melhor dos três porque encontrou o equilíbrio perfeito.

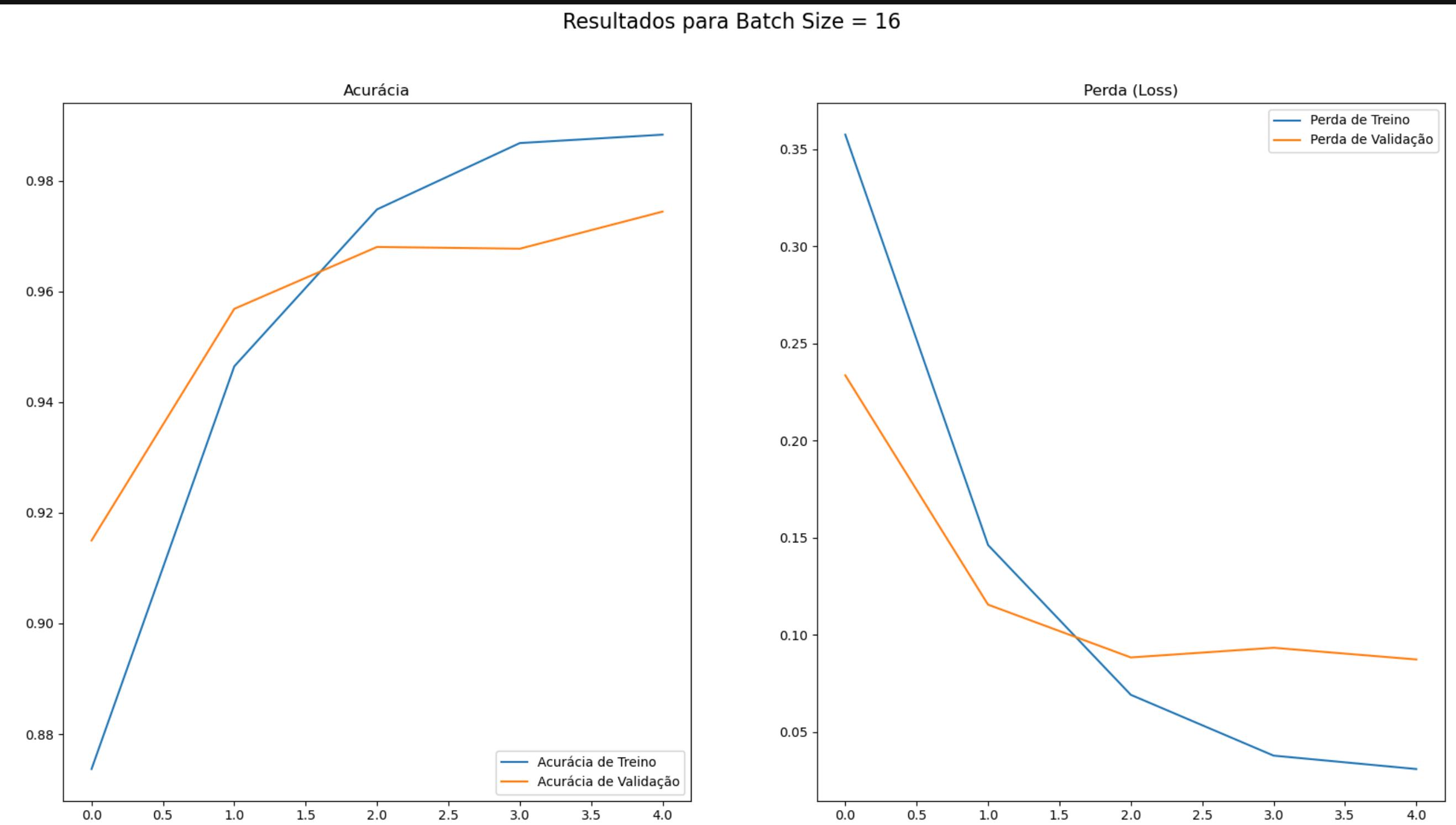


0.0001

- Resultado **Ineficiente**. O modelo está aprendendo de forma estável, mas muito lentamente.
- As curvas de acurácia e perda melhoram devagar, e ele precisaria de muito mais épocas (tempo) para chegar a um bom resultado.

Auto-Tuning

Resultados para Batch Size = 16

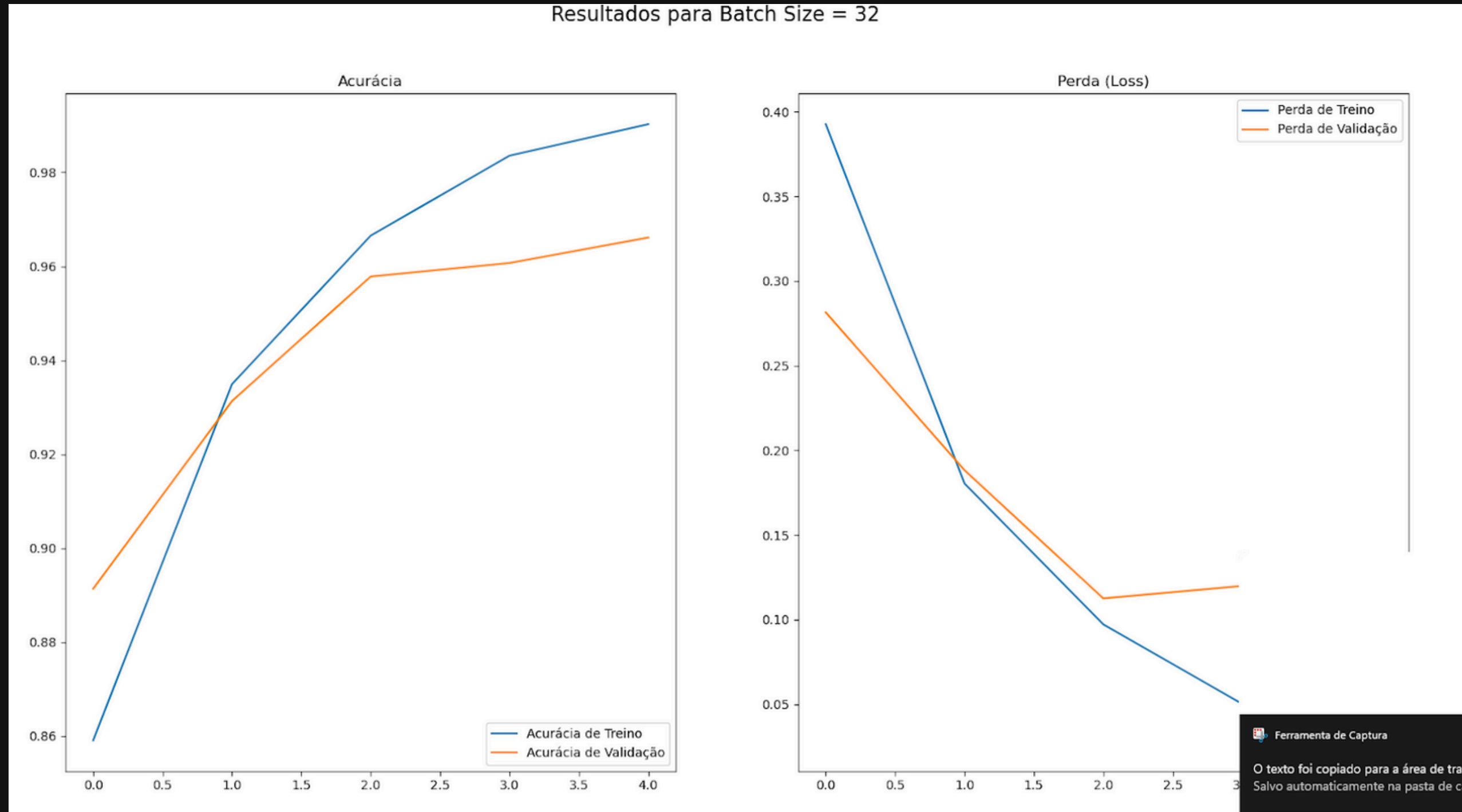


16 = BATCH SIZE

U



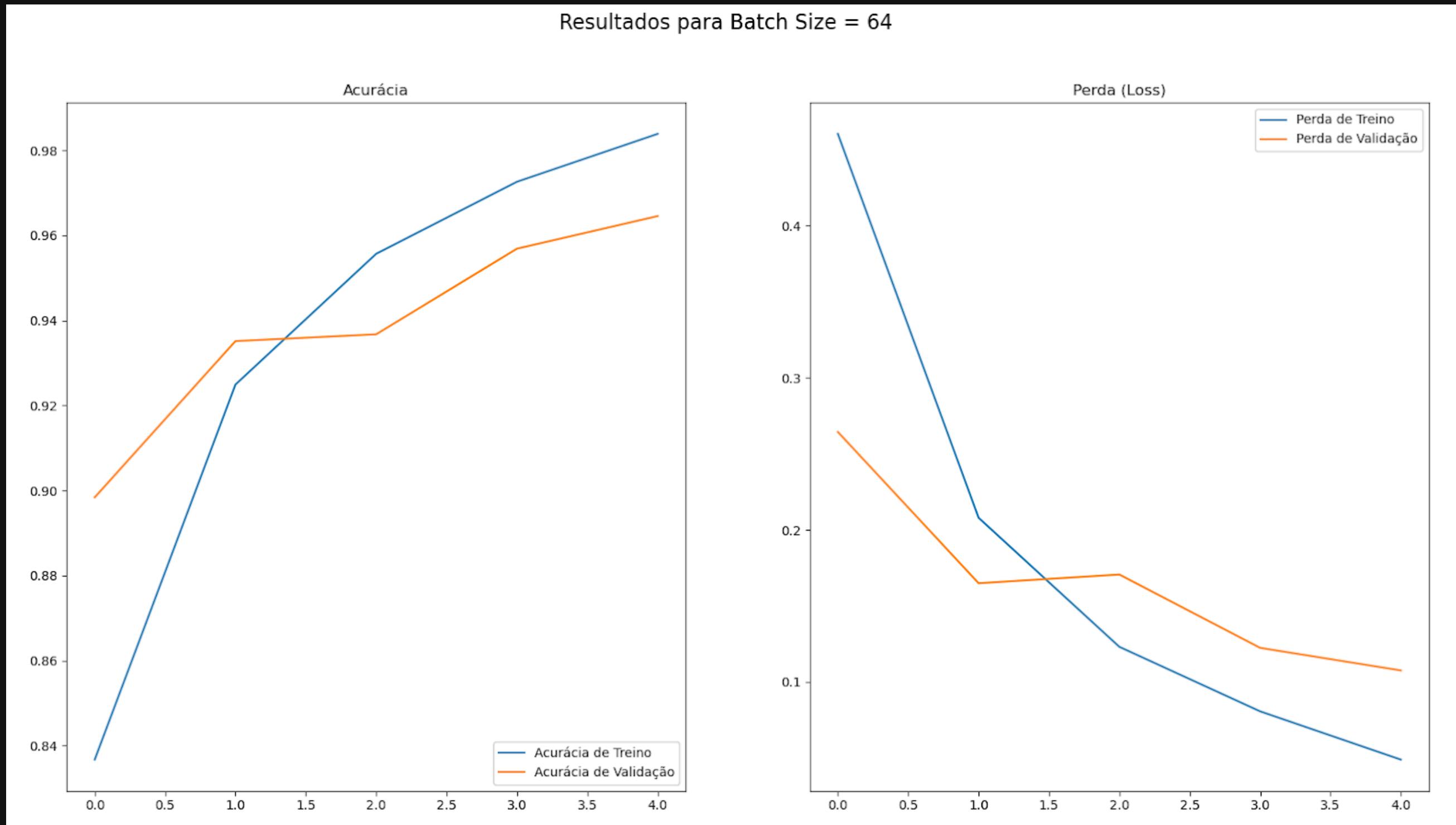
Auto-Tuning



32 = BATCH SIZE

Auto-Tuning

Resultados para Batch Size = 64



64 = BATCH SIZE

BATCH SIZE

16

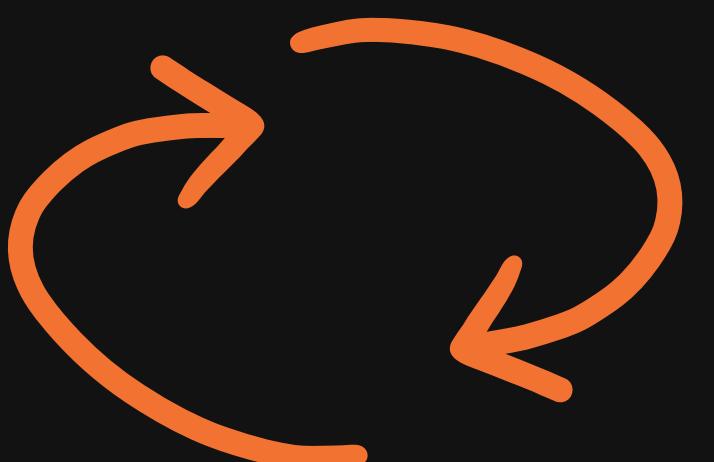
- Foi o **melhor** resultado. Atingiu a **maior** acurácia, a **menor** perda e as curvas de treino e validação ficaram mais próximas.

32

- Foi um **bom** resultado, mas mostrou **instabilidade**. O loss subiu um pouco no meio do treino, o que é um sinal negativo.

64

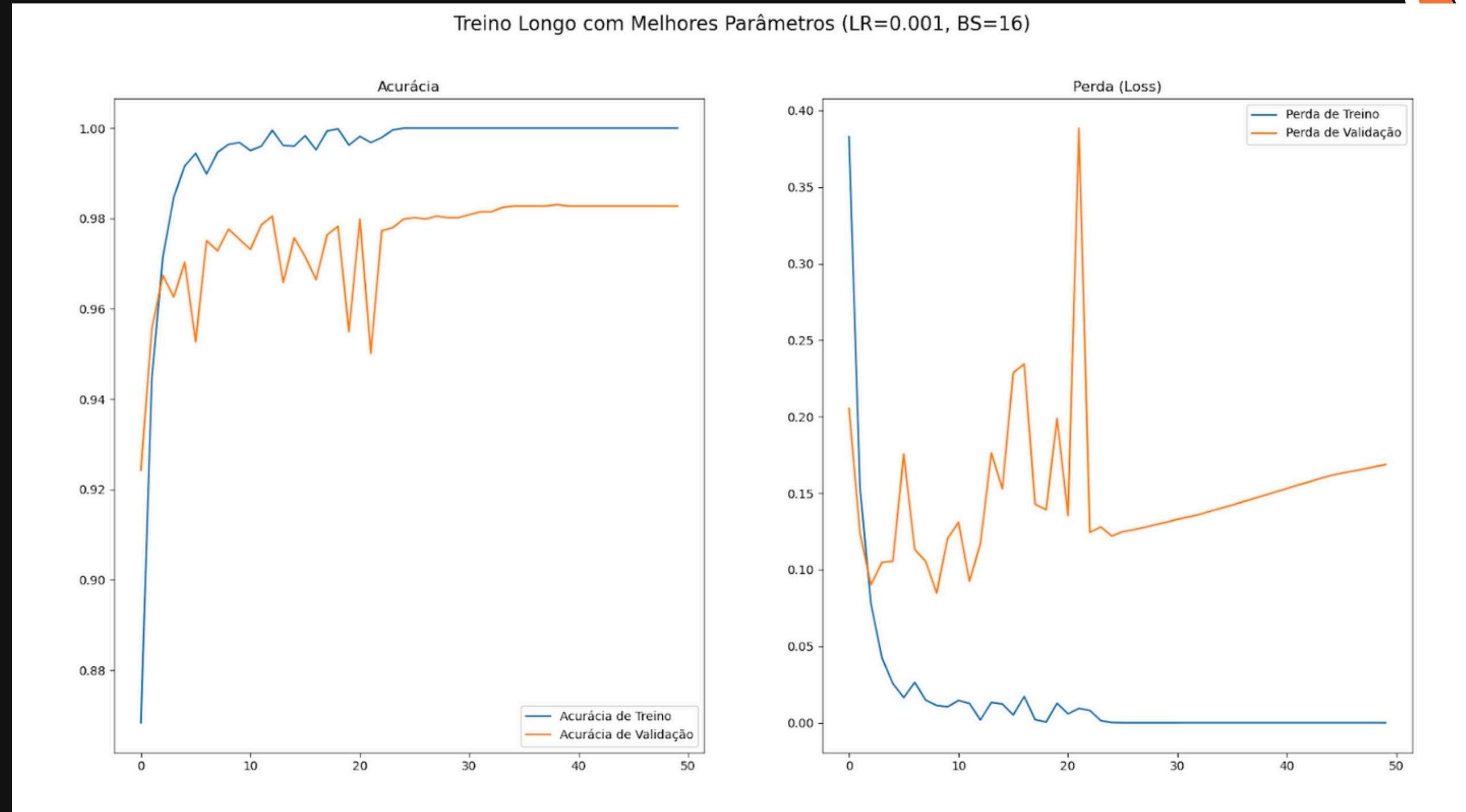
- Foi o resultado mais **fraco**. Apresentou a **menor** acurácia de validação e a **maior** perda entre os três.



Auto-Tuning

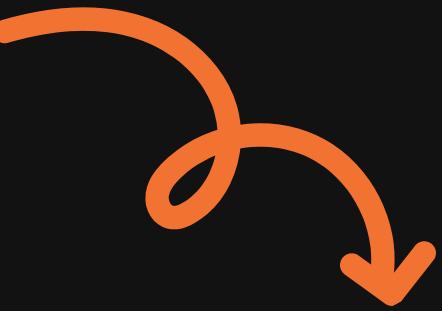


Treino Longo com Melhores Parâmetros (LR=0.001, BS=16)



U

EPOCHS(EPOCAS)



EPOCHS(EPOCAS)

- Com base nos gráficos, a melhor quantidade de épocas é entre **20** e **22** então **escolhemos 21**.
- É **nesse** ponto que o modelo atinge a maior acurácia e a menor perda nos dados de validação
- Após **22** épocas, a perda de validação começa a subir, indicando que o modelo está piorando (overfitting).



EXPERIMENTAÇÃO COMPARATIVA

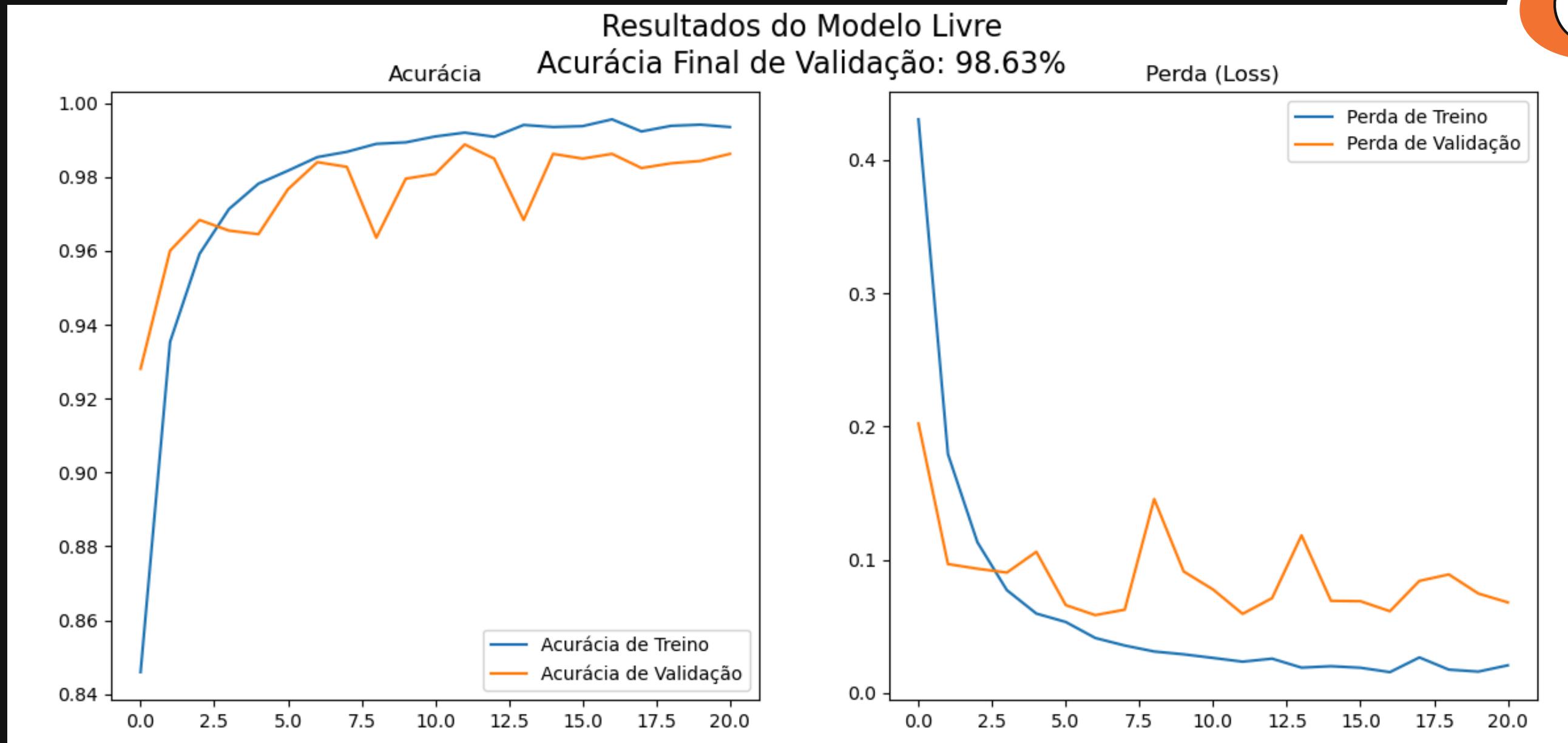
- Com os resultados obtidos por meio do auto-tuning (Taxa de Aprendizagem = 0.001, Batch Size = 16 e 21 épocas), a rede LeNet apresentou uma **melhora** significativa em seu desempenho.
- Onde saiu da taxa de acurácia de **96,13%** para **98,15%**



U

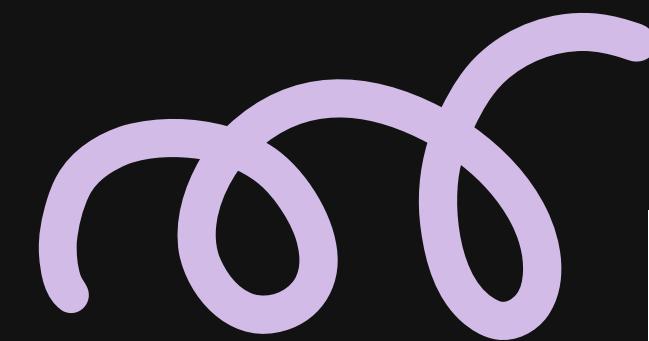
MODELO LIVRE

ok!



A Acurácia Final do modelo livre foi de **98,63%**

U



Forma Gráfica



Camadas do modelo livre;



Formas de ativação;

Dimensão: 96(alt) x 96(larg) x 3(RGB)



Conv2D: extrai características (textura, borda): 96 x 96 x 32

ReLU

Conv2D: Refina e combina as 32 características: 96 x 96 x 32

ReLU

MaxPooling2D: Reduz as dimensões pela metade: 48 x 48 x 32



Conv2D: Extrai 64 características mais complexas: 48 x 48 x 64

ReLU

Conv2D: Refina e combina as 64 características: 48 x 48 x 64

ReLU

MaxPooling2D: Reduz as dimensões novamente: 24 x 24 x 64



Conv2D: Extrai 128 características ainda mais abstratas: 24 x 24 x 128

ReLU

MaxPooling2D: Última redução de dimensionalidade: 12 x 12 x 128



Flatten: Transforma a matriz 12 x 12 x 128 em um vetor único: 18.432 neurônios



Dense: Combina todas as características para iniciar a classificação: 256 neurônios

ReLU

Dropout: Zera 50% dos neurônios (apenas no treino) para evitar overfitting: 256 neurônios



Dense: Gera a probabilidade final para cada classe: 3 neurônios

Softmax

COMPARAÇÃO DE MODELOS



MODELO LIVRE: Acurácia: **98,63%** Loss: **0.1967**

LENET: Acurácia: **96,13%** Loss: **0.1989**

ALEXNET: Acurácia: **95,83%** Loss: **0.1179**





CONCLUSÃO



O Modelo Livre venceu porque sua arquitetura, foi mais eficaz para capturar as características distintas e sutis entre os diferentes tipos de veículos do dataset. Ele conseguiu criar uma representação interna mais rica das imagens, o que resultou em menos erros de classificação, especialmente entre as classes que mais se pareciam. A LeNet, mesmo sendo simples, provou que com os hiperparâmetros corretos, ainda pode ser muito competitiva.

Gabriel Ricetto, João Vitor Pastori e Victor Querino Martins

OBRIGADO

