

Resultado CNN - Frutas

Nome: Gabriel Pellozi Anunciado
Instituição: Universidade Positivo
Localidade: Curitiba, Brasil
RGM: 13423118

Nome: Vinicius Siqueira Bravo
Instituição: Universidade Positivo
Localidade: Curitiba, Brasil
RGM: 8113347900

Resumo — O objetivo deste estudo é comparar a eficiência de três arquiteturas de redes neurais convolucionais (CNN) no treinamento de imagens de dez frutas e avaliar suas métricas de desempenho. São analisadas três arquiteturas diferentes: uma desenvolvida pelos autores deste relatório (Custom), e as outras duas utilizando bibliotecas conhecidas, como DenseNet e ResNet. Através dessa comparação, busca-se identificar qual das arquiteturas apresenta os melhores resultados.

Palavras-chave — CNN, arquiteturas, Custom, Densenet, Resnet, métricas, desempenho, acurácia, perdas, validação.

I. DESCRIÇÃO DAS ARQUITETURAS UTILIZADAS

ResNet: A arquitetura da Residual Network (ResNet) tem como principal objetivo mitigar o problema de degradação durante o treinamento das redes neurais. Por esse motivo, ela apresenta uma maior complexidade em relação à otimização, uma vez que o gradiente tende a diminuir à medida que a rede é treinada. Dentro da estrutura da rede, o valor gerado em uma camada é transmitido adiante para as camadas subsequentes, sem ser alterado. Essa estrutura funciona como um atalho, permitindo uma conexão mais rápida entre a entrada e a saída, sendo a saída a soma resultante desse atalho.

DenseNet: Uma das principais características distintivas da arquitetura DenseNet é a sua conectividade altamente unificada entre as camadas da rede. Essa abordagem promove um compartilhamento eficiente de informações e facilita o treinamento de redes mais profundas. A conexão direta entre todas as camadas permite um fluxo contínuo de informações e preserva os gradientes durante o processo de treinamento.

Custom: A arquitetura Custom foi utilizada para estruturar o CNN de forma sequencial, permitindo a adição de outras camadas para a geração da rede neural. Para o processamento de imagens, empregamos a camada Conv2D, que realiza convoluções em duas dimensões, utilizando altura e largura como valores principais. Com o objetivo de otimizar a camada e agilizar seu treinamento/compilação, aplicamos a (1) função Flatten para converter os dados bidimensionais em um vetor unidimensional. Esse formato é adequado para o uso de camadas densas (Dense). O cálculo é realizado da seguinte forma:

$$X = (\text{tamanho_lote}, \text{altura} * \text{largura} * \text{canais_cores}) \quad (1)$$

II. DESCRIÇÃO DO DATASET

O relatório utiliza uma amostra pequena de frutas de um conjunto de dados obtido no [Kaggle](#). O objetivo é treinar um modelo para classificar dez tipos de frutas: Açaí, Abacate, Banana, Cereja, Café, Guaraná, Limão, Abacaxi, Morango e Melancia.

O Google Colab foi utilizado como ferramenta para teste e validação do modelo. O conjunto de dados consiste em um total de 8140 imagens, distribuídas em pastas contendo entre 600 e 1000 imagens cada uma. Essa variação entre as pastas permite testar a qualidade do modelo de CNN em um conjunto maior de dados.

Foram explorados vários modelos de CNN para classificar frutas que não estão presentes no conjunto de dados original. O objetivo foi encontrar o melhor modelo a ser utilizado para uma aplicação real. Com conjunto de dados sendo dividido em 70% para treinar o modelo e 30% para validar o desempenho do modelo.

III. MÉTODO

O projeto foi desenvolvido utilizando os seguintes métodos:

- Treinamento de três classificadores CNN, cada um com três variações, totalizando nove modelos para validação. Essa abordagem teve o propósito de testar e compreender melhor o uso de algoritmos específicos, como Binarizador ou Categórico.
- Análise de desempenho utilizando o perceptron para identificar se o modelo apresenta sinais de overfitting, underfitting ou ajuste ideal. Com objetivo para encontrar o melhor modelo dentre todos os gerados.
- Salvar modelos treinados para classificar até 10 tipos de frutas, permitindo a geração rápida de classificações precisas para diferentes imagens. Essa abordagem economiza tempo e recursos, evitando a necessidade de treinamento constante. É uma técnica eficiente e confiável para a classificação de imagens de frutas.

IV. RESULTADOS

ResNet

Adam sem hiperparâmetro: Treinamento do classificador em vinte épocas, começando com 35% de acurácia e terminando com 56.86%, com testes tendo valor de 57.34%, já os valores de perda bem alto, finalizando com valor de 1.32

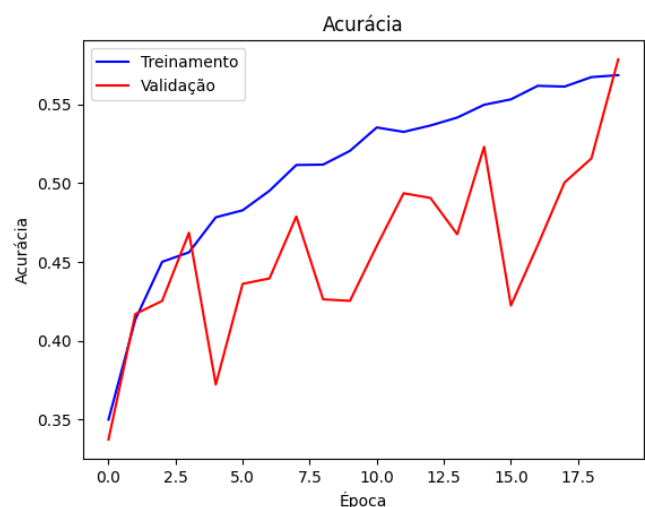


Fig 1 ResNet - Adam Sem Hiperparâmetro - Acurácia

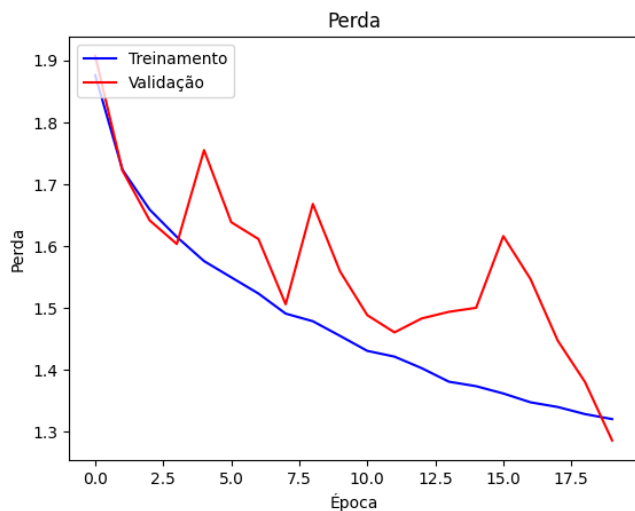


Fig 2 ResNet - Adam Sem Hiperparâmetro - Perda

Esse modelo demonstra ter um caso de underfitting, pois sua acurácia está bem baixa, com a acurácia de validação sendo inconsistente durante a geração de épocas, no final o treino e teste se aproximando em seus valores, isso sendo representado também pela perda, tanto a acurácia e perda possui aumente quanto mais épocas foram geradas.

Aplicando testes em imagens fora do DataSet, em três imagens, elas sendo banana, cereja e melancia, este modelo resultou em dois acertos e um erro, mostrando que mesmo com uma acurácia baixa, conseguiu encontrar boa parte das imagens.

Descida de gradiente estocástico (SGD): Treinamento do classificador em vinte épocas, começando com 10.6% de acurácia e terminando com 34%, com testes tendo valor de 35%, já os valores de perda bem alto, finalizando com valor de 1.92

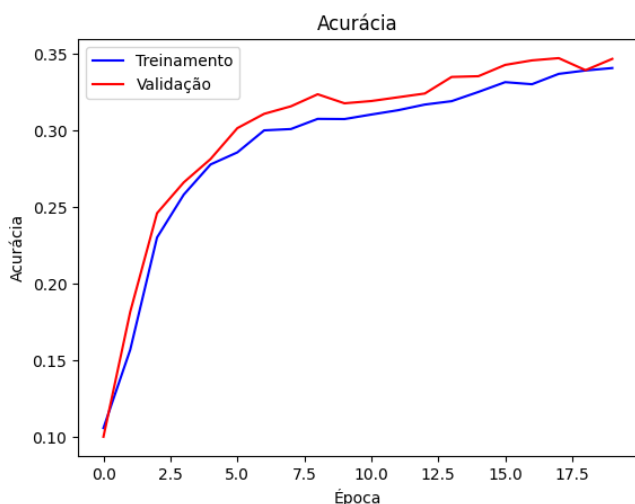


Fig 3 ResNet - SGD - Acurácia

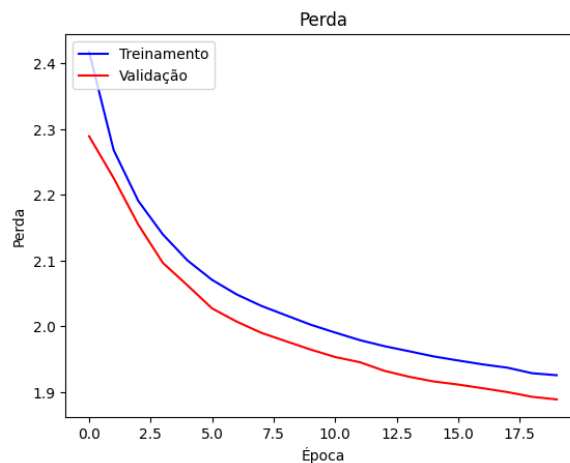


Fig 4 ResNet - SGD - Perda

O resultado desse modelo está próximo de um best fit, porém a acurácia é bem baixa e o valor de perda é alto, o que resulta em valores errados para quando for classificar uma imagem fora do banco de dados, aplicando testes em imagens fora do DataSet, em três imagens, elas sendo banana, cereja e melancia, este modelo resultou em um acerto.

Adam com hiperparâmetro: Treinamento do classificador em vinte épocas, começando com 28% de acurácia e terminando com 48.8%, com testes tendo valor de 43%, já os valores de perda terminou com valor de 1.59

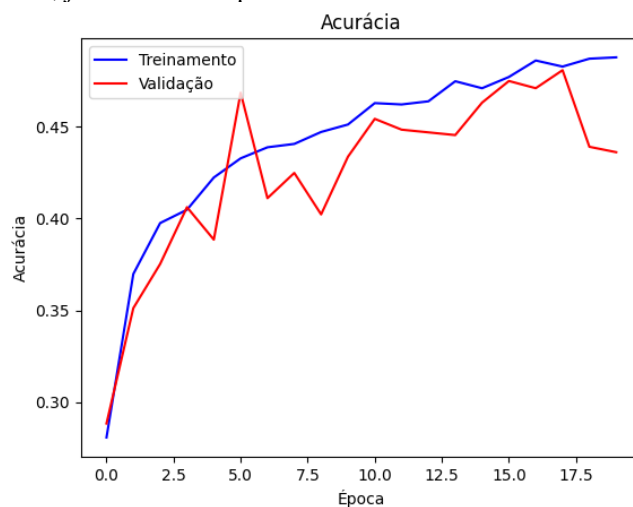


Fig 5 ResNet - Adam Com Hiperparâmetro - Acurácia

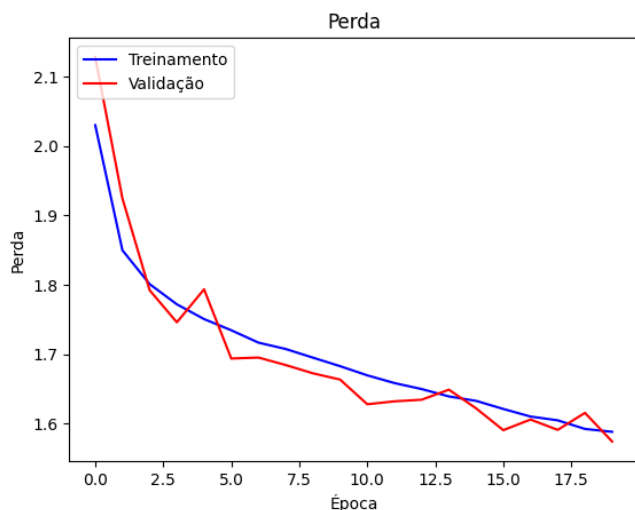


Fig 6 ResNet – Adam Com Hiperparâmetro - Perda

Esse modelo apresenta um caso de best fit evidente, durante a geração de épocas, a acurácia de treinamento e validação se mostra próximas, o que também é observado na perda. Tanto a acurácia quanto a perda aumentam à medida que mais épocas são geradas.

Ao realizar testes em imagens que não fazem parte do conjunto de dados, utilizando três exemplos: banana, cereja e melancia, o modelo obteve dois acertos e um erro. Esses resultados demonstram que, apesar da acurácia baixa, o modelo foi capaz de reconhecer a maioria das imagens corretamente.

DenseNet

Adam sem hiperparâmetro: Treinamento do classificador em vinte épocas, começando com 50.9% de acurácia e terminando com 96%, com testes tendo valor de 75%, já os valores de perda são bem baixos, concluindo com valor de 0.02

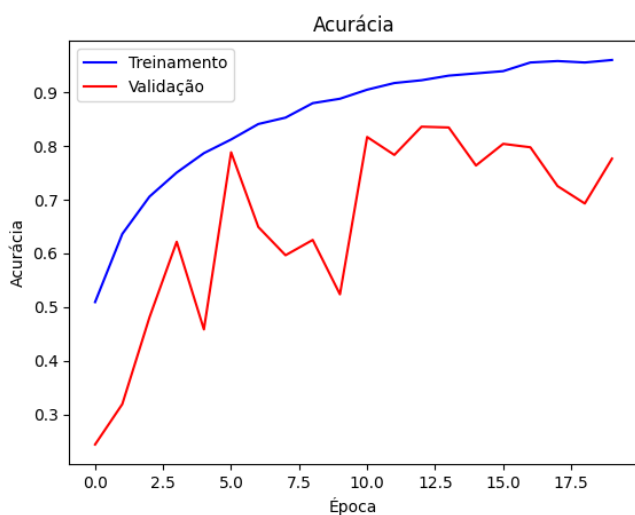


Fig 7 DenseNet – Adam Sem Hiperparâmetro - Acurácia

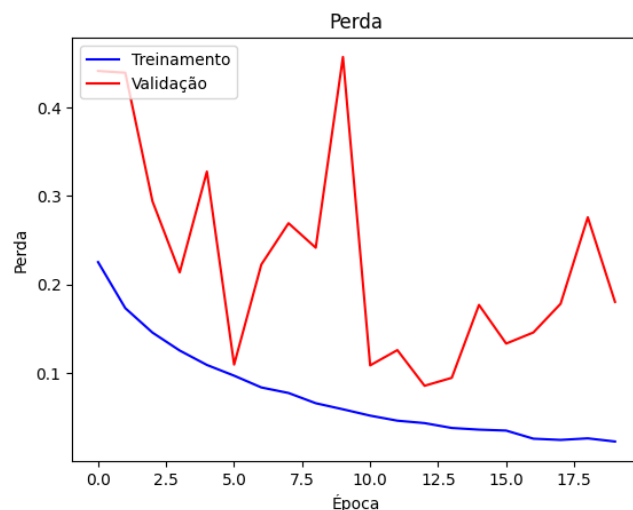


Fig 8 DenseNet – Adam Sem Hiperparâmetro - Perda

Esse modelo durante seu treinamento gerou overfitting, mesmo demonstrando uma acurácia alta, o resultado de validação resulta em grandes variações ao passar das épocas, mostrando uma inconsistência, ele se aplica para perda dos dados também, esse é um dos melhores modelos gerado para classificar esse conjunto de imagens.

Contudo, ao conduzir testes em imagens não incluídas no conjunto de dados, utilizando três exemplos (banana, cereja e melancia), o modelo registrou dois acertos e um erro. Esses resultados revelam que, embora sua acurácia seja alta, o modelo conseguiu reconhecer corretamente certas imagens.

Descida de gradiente estocástico (SGD):

Treinamento do classificador em vinte épocas, começando com 10% de acurácia e terminando com 32%, com testes tendo valor similares com de treinamento, já os valores de perda é baixo, chegando no valor de 0.42, com perdas de testes sendo similares com treinamento

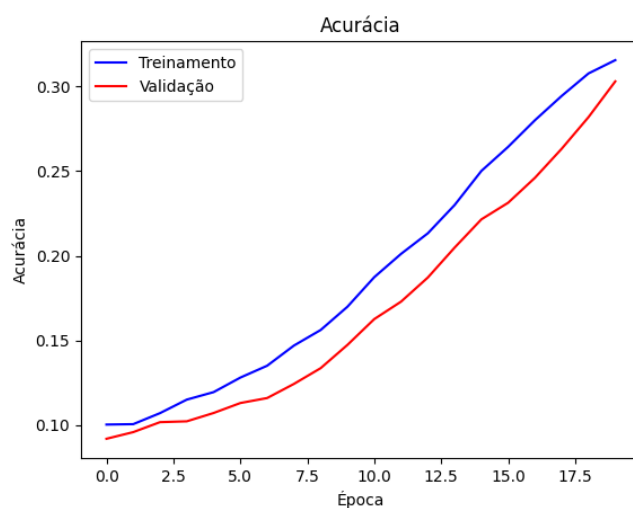


Fig 9 DenseNet – SGD - Acurácia

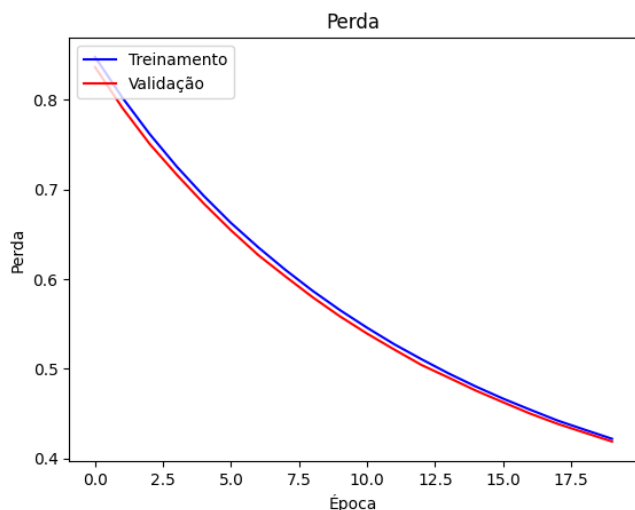


Fig 10 DenseNet – SGD - Perda

O desempenho atual desse modelo está próximo de um ajuste ótimo, porém, ele produz valores incorretos ao classificar imagens que estão fora do banco de dados. A perda do modelo é baixa quando aplicado a testes com três imagens específicas - banana, cereja e melancia - resultando em nenhuma classificação correta. Isso indica que o modelo pode ser aprimorado por meio da geração de mais épocas de treinamento.

Adam com hiperparâmetro: Treinamento do classificador em vinte épocas, começando com 53.3% de acurácia e terminando com 98%, com testes tendo valor de 75%, já os valores de perda é baixo, com valores de treino com 0.01 com testes chegando nos 0.2

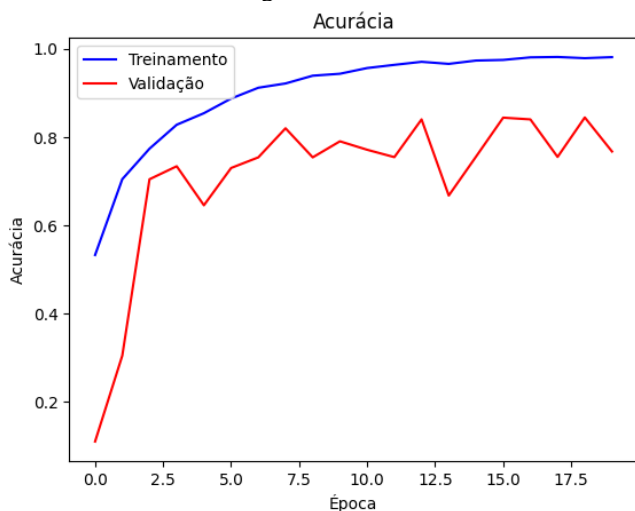


Fig 11 DenseNet – Adam Com Hiperparâmetro - Acurácia

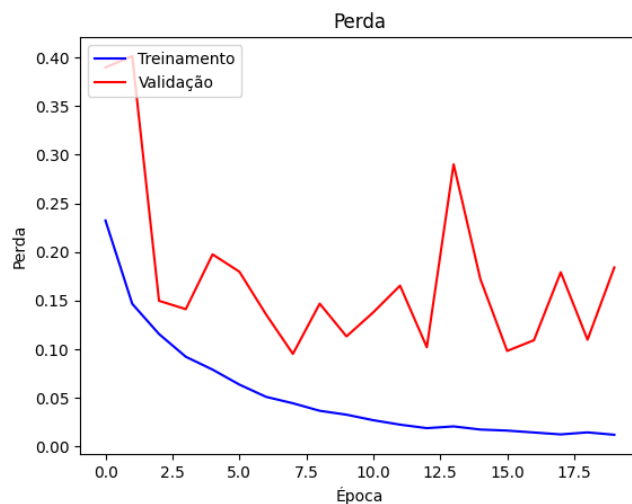


Fig 12 DenseNet – Adam Com Hiperparâmetro - Perda

Durante o treinamento, este modelo apresentou sinais de overfitting, o que se refletiu em uma alta acurácia, mas resultou em grandes variações nos resultados de validação ao longo das épocas. Essa inconsistência também se manifesta na perda dos dados.

No entanto, ao realizar testes em imagens que não foram incluídas no conjunto de dados, utilizando exemplos como banana, cereja e melancia, o modelo classificou todas as imagens de forma incorreta. Esses resultados revelam que, embora a acurácia seja alta, o modelo não conseguiu reconhecer corretamente nenhuma das imagens, gerando apenas resultados falsos positivos.

Custom

Adam sem hiperparâmetro: Treinamento do classificador em vinte épocas, começando com 42% de acurácia e terminando com 99%, com testes tendo valor de 60%, já os valores de perda bem baixo para treino, contendo valor final de 0.03, porém para testes o valor cresceu com tendo resultado final sendo 4

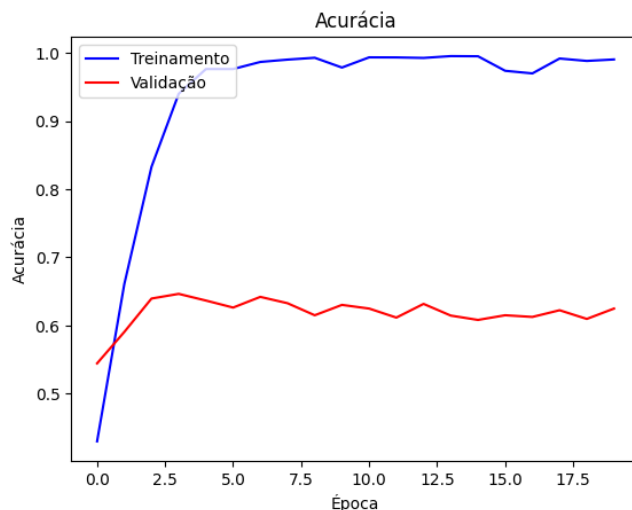


Fig 13 Custom – Adam Sem Hiperparâmetro - Acurácia

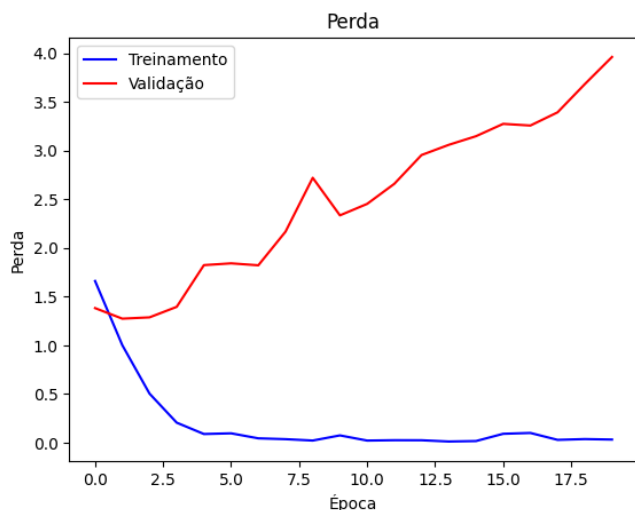


Fig 14 Custom – Adam Sem Hiperparâmetro - Perda

Durante o treinamento deste modelo, foi observado um problema de overfitting, apesar da demonstração de alta acurácia. Os resultados de validação revelaram valores inferiores, indicando uma inconsistência. Esse problema também se manifestou na perda de dados, onde os valores de perda nos testes foram significativamente mais elevados. Apesar dessas questões, este modelo ainda se destaca como um dos melhores para classificar esse conjunto de imagens.

No entanto, ao realizar testes em imagens que não faziam parte do conjunto de dados, utilizando três exemplos (banana, cereja e melancia), o modelo apresentou dois acertos e um erro. Esses resultados mostram que, embora sua acurácia seja alta, o modelo ainda é capaz de reconhecer corretamente algumas imagens.

Descida de gradiente estocástico (SGD):

Treinamento do classificador em vinte épocas, começando com 13.9% de acurácia e terminando com 46.9%, com testes tendo valor de 43%, valores são grandes com resultado de 1.63 para treino, porém com valores de teste próximos sendo 1.65

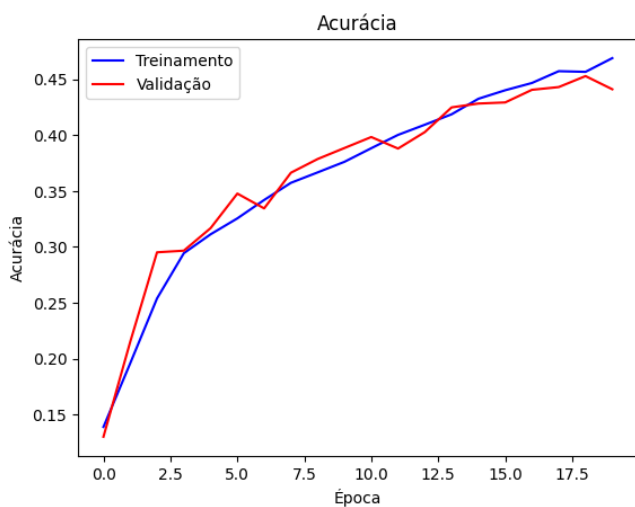


Fig 15 Custom – SGD - Acurácia

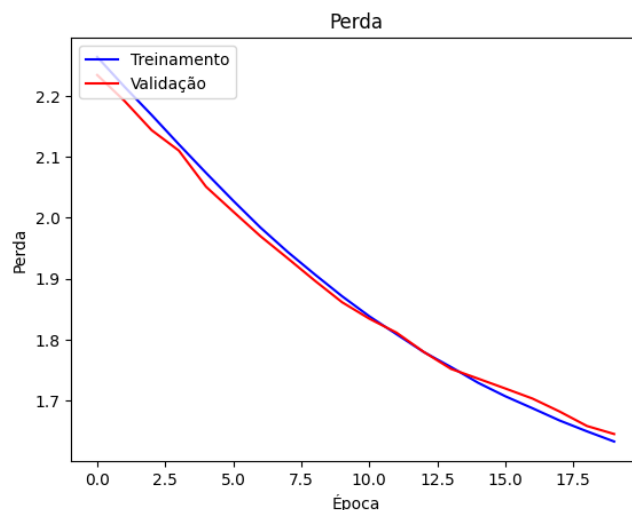


Fig 16 Custom – SGD - Perda

O desempenho atual deste modelo está quase alcançando um ajuste ótimo. No entanto, ele apresenta dificuldades ao classificar imagens que não fazem parte do banco de dados, produzindo resultados incorretos. A perda do modelo é baixa quando testado com três imagens específicas - banana, cereja e melancia - resultando em classificações corretas. Isso sugere que o modelo pode ser aprimorado por meio da realização de mais épocas de treinamento.

Adam com hiperparâmetro: Treinamento do classificador em vinte épocas, começando com 33.4% de acurácia e terminando com 99.7%, com testes tendo valor de 61%, já os valores de perda é menor para treinos e alto durante testes, com valores respectivo para 0.05 de treino e 1.26 para testes

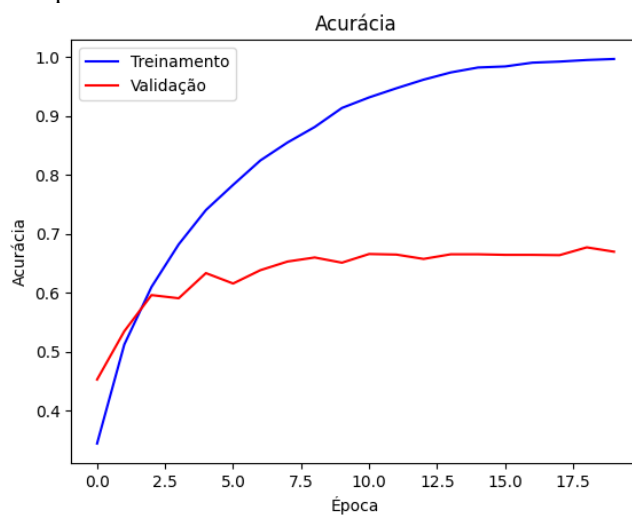


Fig 17 Custom – Adam Com Hiperparâmetro - Acurácia

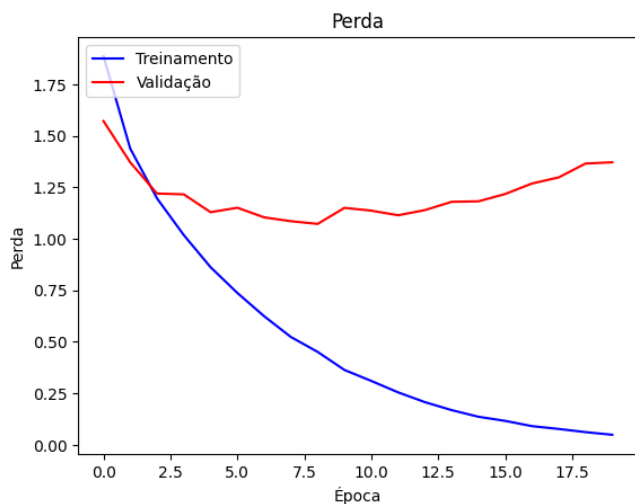


Fig 18 Custom – Adam Com Hiperparâmetro - Perda

Durante o treinamento deste modelo, identificamos um problema de overfitting, apesar da alta acurácia observada na demonstração. Os resultados de validação revelaram valores inferiores, evidenciando uma inconsistência. Esse problema também afetou a perda de dados, com valores significativamente mais elevados nos testes. Apesar dessas questões, este modelo continua sendo um dos melhores para classificar este conjunto de imagens.

No entanto, ao testarmos imagens que não faziam parte do conjunto de dados, utilizando três exemplos (banana, cereja e melancia), o modelo obteve dois acertos e um erro. Esses resultados mostram que, embora sua acurácia seja alta, o modelo ainda possui algumas limitações na correta identificação de certas imagens.

V. CONCLUSÃO

O perceptron é um algoritmo fundamental no campo do aprendizado de máquina e das redes neurais. Embora possua algumas limitações, sua evolução resultou no desenvolvimento de redes neurais mais complexas, capazes de resolver problemas mais desafiadores. O perceptron ressaltou a importância dos pesos, da função de ativação e do ajuste iterativo para otimizar o desempenho do modelo. Sua aplicação abrange diversas áreas, incluindo visão computacional e processamento de linguagem natural.

A análise comparativa dos diferentes modelos revelou resultados variados, evidenciando a necessidade de uma análise mais aprofundada para determinar qual modelo obteve o melhor desempenho em termos de geração de épocas. O DenseNet se destacou nesse aspecto, mantendo seus hiperparâmetros inalterados e alcançando uma alta acurácia. Esse modelo obteve sucesso na identificação de dois terços das imagens desconhecidas, demonstrando sua eficácia. No entanto, é importante mencionar o potencial de outros modelos, como o ResNet, que mesmo com uma acurácia um pouco menor, se aproximou do melhor ajuste e identificou duas imagens durante a validação.

O estudo do perceptron e sua evolução são essenciais para aprimorar as redes neurais e o aprendizado de máquina. O uso de modelos como o DenseNet e o ResNet mostra a importância de encontrar o equilíbrio entre acurácia e eficiência computacional, permitindo que essas redes resolvam problemas complexos em diferentes domínios. À

medida que a pesquisa e o desenvolvimento continuam, espera-se que novos avanços sejam alcançados, impulsionando ainda mais o campo da inteligência artificial.