

Equipe 3 - Fruit Classification

Alunos:

Fábio Carvalho Simões

Lívia Braga Sydrião de Alencar

Micael José de Lima

Nicolas Kalebe Menezes da Silva

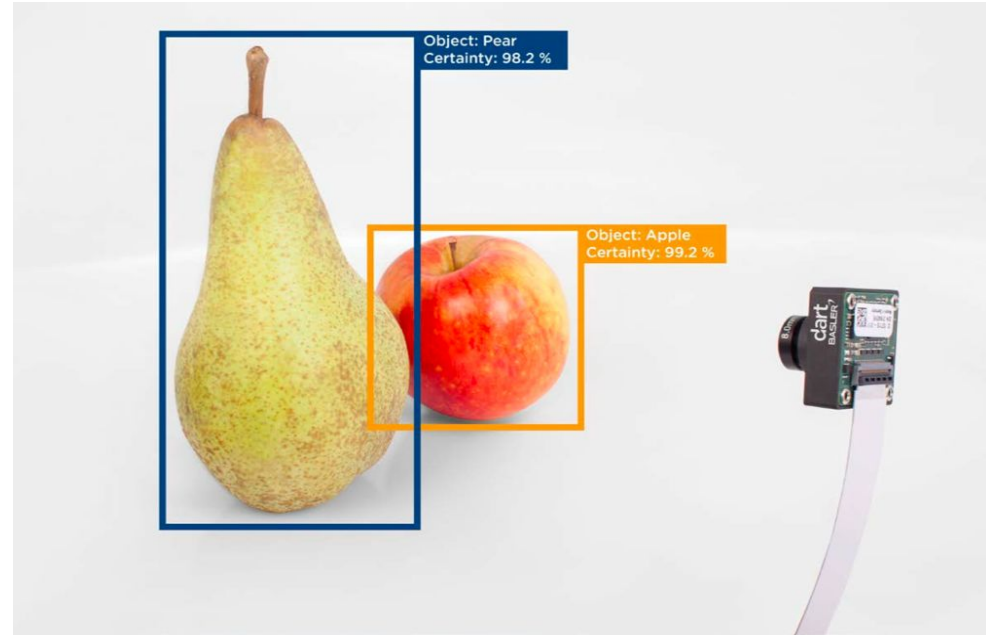
Rafael de Oliveira Feitosa

Yuri Ramos Ribeiro



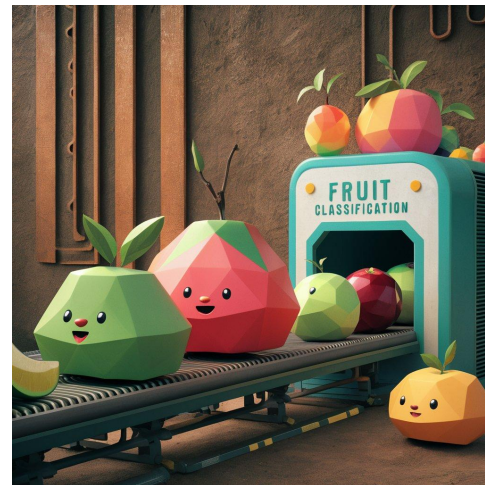
Sumário

- Introdução
- Metodologia
- Pré-processamento
- Modelos
- Avaliação
- Predição
- Conclusão
- Referências



Introdução

- Realização de um projeto de classificação de imagens de diferentes classes de frutas utilizando a biblioteca Tensor Flow.
- Divisão do processo em várias etapas:
 - Pré-processamento dos dados.
 - Exploração das funcionalidades do Tensor Flow.
 - Construção de modelos de aprendizado de máquina.
 - Avaliação dos modelos por diferentes métricas.
 - Predição das imagens.



Introdução

- Importância da classificação automática de imagens de frutas em áreas como:
 - Agricultura.
 - Comércio.
 - Alimentação.
- Desafios da tarefa de classificação:
 - variabilidade nas formas, cores e texturas das frutas.
 - Condições de iluminação e fundo das imagens.
- Importância da solução:
 - Aumento da eficiência e redução dos custos operacionais em setores como agricultura e varejo.
 - Melhoria na qualidade dos produtos entregues ao consumidor.
 - Contribuição para sistemas de monitoramento e análise agrícola, fornecendo dados valiosos para a tomada de decisões.



Metodologia

Base de dados:

Número total de imagens:
22495 imagens

Número de Classes:
33

Tamanho da imagem:
100x100

OBS: Base de dados reduzido

Método:

3 modelos diferentes de
CNN



Pré-Processamento

- Verificando valores nulos e duplicatas:
 - Resultado: ok
- Dataframe e Redução do tamanho do dataset:
 - Validação, Treinamento e Teste (fração de 88%)
- Carregando imagem e criando um array de dados:
 - Validação Treinamento: (14832, 100, 100, 3)
 - Teste: (4964, 100, 100,3)
- Codificação das classes em valores numéricos
- Divisão de Treinamento, Validação e Teste:
 - Treinamento: (11865, 100, 100, 3)
 - Validação: (2967, 100, 100, 3)
 - Teste: (4964,100,100,3)
- Aplicação da função prepare:
 - Normalização dos dados (1/ 255)
 - Batch_size = 128
 - shuffle = True
 - Augment = True

Modelo 1 – PRINCIPAL

Definição do Modelo CNN:

- Define um modelo sequencial
- Adiciona uma camada convolucional
 - filters = 5
 - kernel_size = 3
 - activation = “relu”
 - input_shape = (100, 100, 3)
- Adiciona uma camada de pooling Máximo
 - pool_size = 2
 - padding = ‘valid’
- Transformação dos dados para camada densa
- Adiciona uma camada densa(totalmente conectada)
 - len(classes)
 - activation = ‘softmax’

Compilação do modelo:

- Configura para treinamento
 - loss = “categorical_crossentropy”
 - optimizer= “Adam”
 - metrics = [“Accuracy”]

Modelo 2

Definição do Modelo CNN:

- Define um modelo sequencial
- Adiciona uma camada convolucional
 - filters = 10
 - kernel_size = 5
 - activation = "selu"
 - input_shape = (100, 100, 3)
- Adiciona uma camada de pooling Máximo
 - pool_size = 4
 - padding = 'same'
- Transformação dos dados para camada densa
- Adiciona uma camada densa(totalmente conectada)
 - len(classes)
 - activation = 'softmax'

Compilação do modelo:

- Configura para treinamento
 - loss = "categorical_crossentropy"
 - optimizer= "SGD"
 - metrics = ["Accuracy"]

Modelo 3

Definição do Modelo CNN:

- Define um modelo sequencial
- Adiciona uma camada convolucional
 - filters = 15
 - kernel_size = 7
 - activation = "leaky_relu"
 - input_shape = (100, 100, 3)
- Adiciona uma camada de pooling Máximo
 - pool_size = 6
 - padding = 'valid'
- Transformação dos dados para camada densa
- Adiciona uma camada densa(totalmente conectada)
 - len(classes)
 - activation = 'softmax'

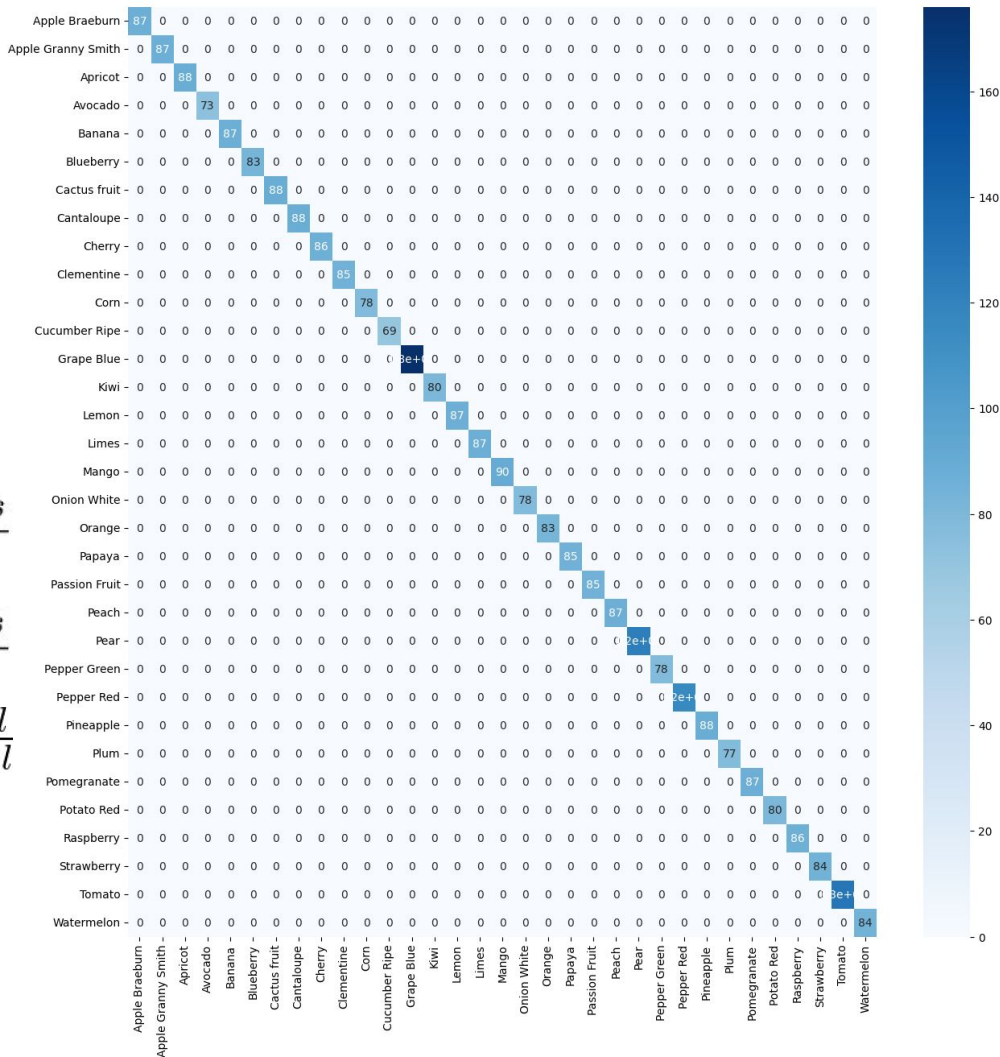
Compilação do modelo:

- Configura para treinamento
 - loss = "categorical_crossentropy"
 - optimizer= "Adam"
 - metrics = ["Accuracy"]

$$Precision = \frac{Previsões\ Positivas\ Corretas}{Previsões\ Positivas}$$

$$F1\ Score = 2 * \frac{Precisão * Recall}{Precisão + Recall}$$

↗ Precisão: 1.0 ↗ Recall: 1.0 ↗ F1: 1.0



Avaliação: Matriz de Confusão – Modelo 2

Cada célula da matriz representa o número de instâncias de uma classe prevista correta ou incorretamente pelo modelo.

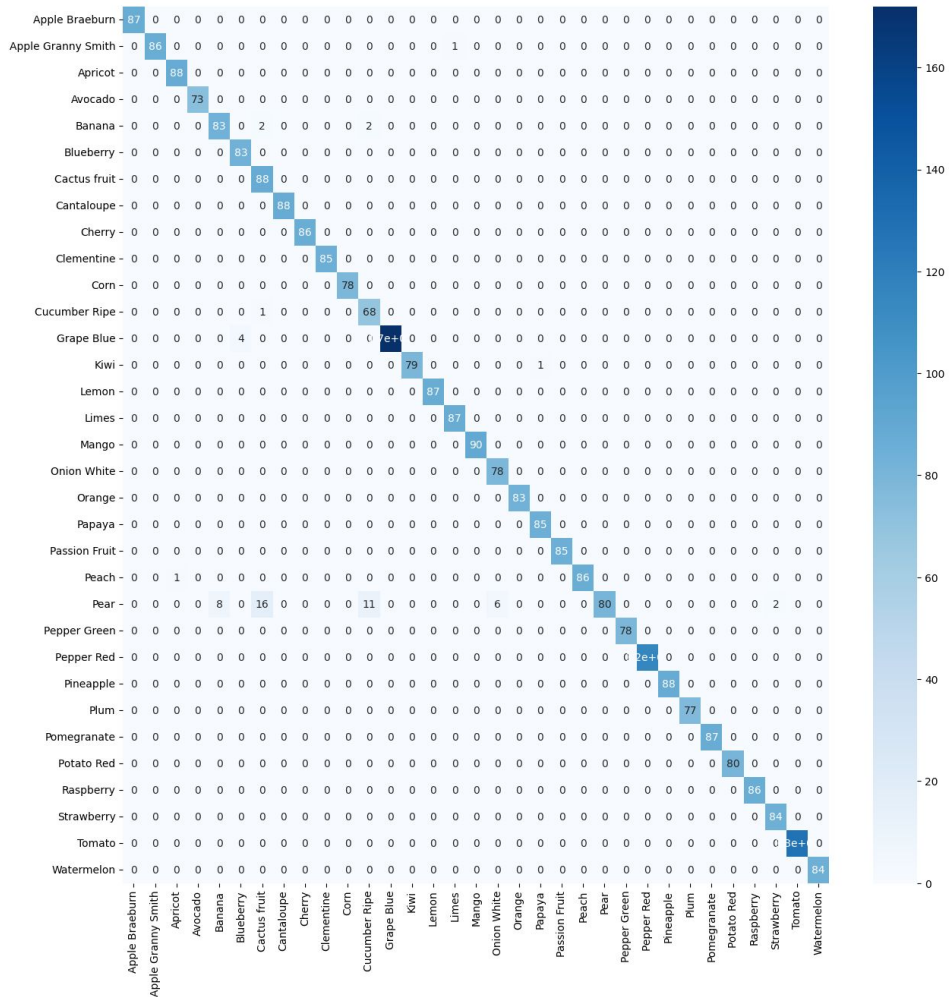
$$Precision = \frac{Previsões\ Positivas\ Corretas}{Previsões\ Positivas}$$

$$Recall = \frac{Previs\tilde{o}es\ Positivas\ Corretas}{Exemplos\ Positivos}$$

$$F1\ Score = 2 * \frac{Precisão * Recall}{Precisão + Recall}$$

Pontuação:

➡ Precisão: 0.98 ➡ Recall: 0.99 ➡ F1: 0.98



Avaliação: Matriz de Confusão – Modelo 3

Cada célula da matriz representa o número de instâncias de uma classe prevista correta ou incorretamente pelo modelo.

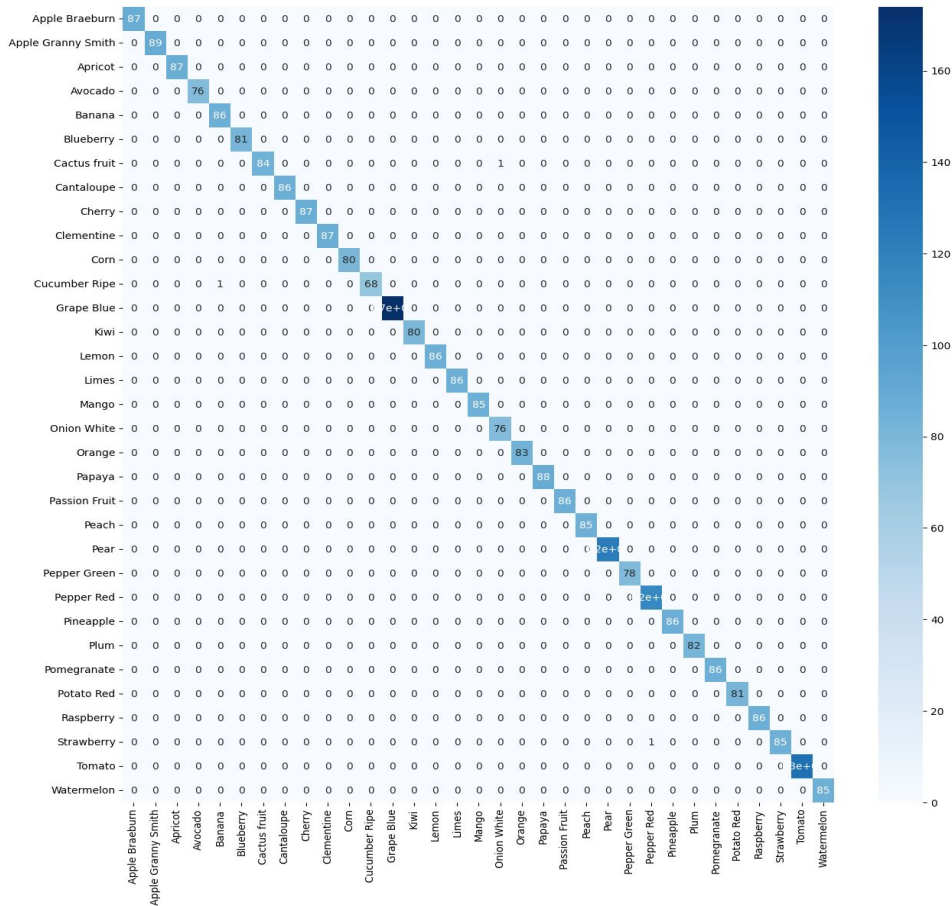
$$Precision = \frac{Previsões\ Positivas\ Corretas}{Previsões\ Positivas}$$

$$Recall = \frac{Previs\tilde{o}es\ Positivas\ Corretas}{Exemplos\ Positivos}$$

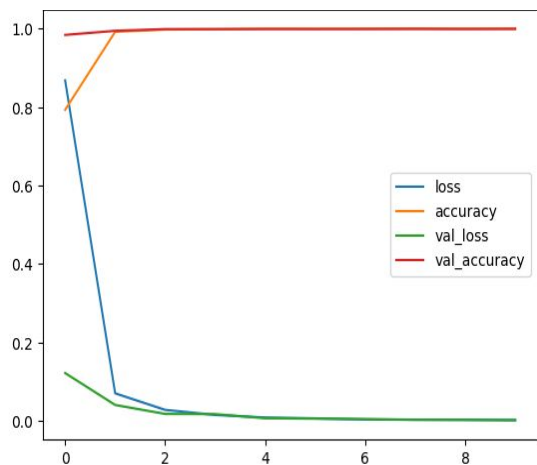
$$F1\ Score = 2 * \frac{Precisão * Recall}{Precisão + Recall}$$

Pontuação:

➡ Precisão: 0.99 ➡ Recall: 0.99 ➡ F1: 0.99

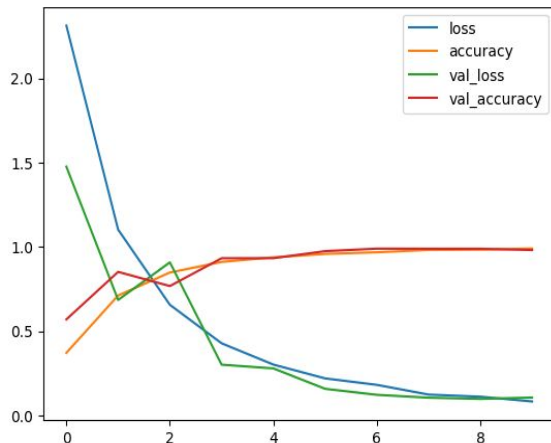


Avaliação: Accuracy x Loss



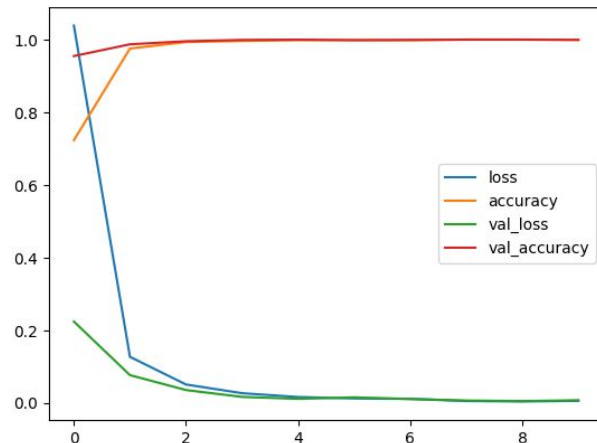
Modelo 1

- 10 epochs
- loss : 0.0
- val_loss: 0.0
- accuracy: 1.0
- val_acurracy: 1.0



Modelo 2

- 10 epochs
- loss : 0.0
- val_loss: 0.0
- accuracy: 1.0
- val_acurracy: 1.0



Modelo 3

- 10 epochs
- loss : 0.0
- val_loss: 0.0
- accuracy: 1.0
- val_acurracy: 1.0

Avaliação: TP, FP, FN e TN

	Previsão: Sim	Previsão: Não
Realidade: Sim	Positivo Verdadeiro	Falso Negativo
Realidade: Não	Falso Positivo	Negativo Verdadeiro

- Verdadeiros Positivos (TP)
- Falsos Negativos (FN)
- Falsos Positivos (FP)
- Verdadeiros Negativos (TN)

Modelo 1	Modelo 2	Modelo 3
⇒ TP = 87	⇒ TP = 86	⇒ TP = 87
⇒ FN = 0	⇒ FN = 54	⇒ FN = 3
⇒ FP = 0	⇒ FP = 1	⇒ FP = 0
⇒ TN = 2880	⇒ TN = 2881	⇒ TN = 2875

Avaliação: Sensibilidade, Especificidade e Média Geométrica

G-mean indica o equilíbrio entre o desempenho na classe majoritária e minoritária e leva em consideração tanto a sensibilidade quanto a especificidade.

- Sensibilidade: Recall
- Especificidade: $1 - (\text{False Positive} / (\text{False Negative} + \text{True Negative}))$
- G-Means = $\sqrt{\text{Sensibilidade} * \text{Especificidade}}$

Modelo 1

- ⇒ Sensibilidade = 1.0
- ⇒ Especificidade = 1.0
- ⇒ G-mean = 1.0

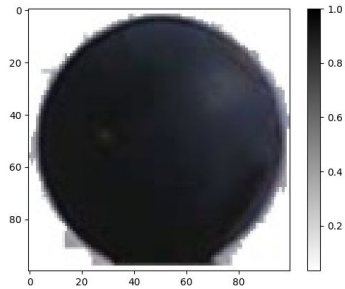
Modelo 2

- ⇒ Sensibilidade = 0.61
- ⇒ Especificidade = 0.98
- ⇒ G-mean = 0.77

Modelo 3

- ⇒ Sensibilidade = 0.96
- ⇒ Especificidade = 0.99
- ⇒ G-mean = 0.98

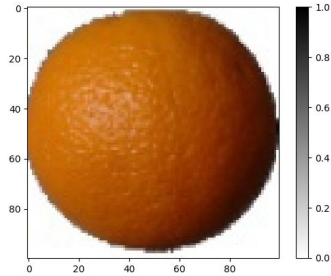
Predição: Modelo 1



Predição 1

⇨ Esperada:
Grape Blue

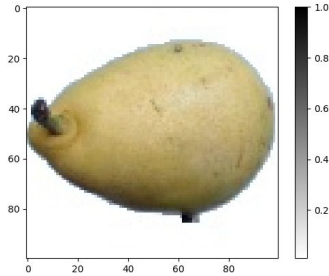
⇨ Predição:
Grape Blue



Predição 2

⇨ Esperada:
Orange

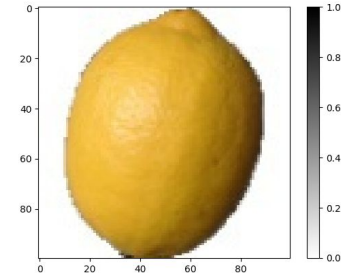
⇨ Predição:
Orange



Predição 3

⇨ Esperada:
Pear

⇨ Predição:
Pear

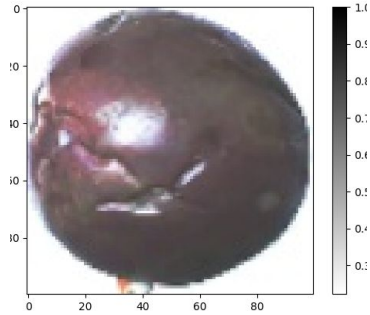


Predição 4

⇨ Esperada:
Lemon

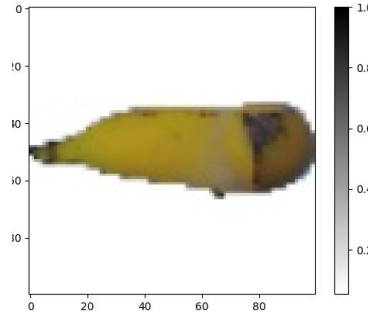
⇨ Predição:
Lemon

Predição: Modelo 2



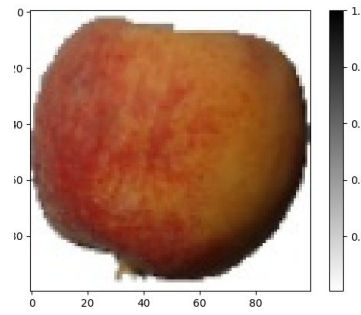
Predição 1

⇨ Esperada:
Passion Fruit
⇨ Predição:
Passion Fruit



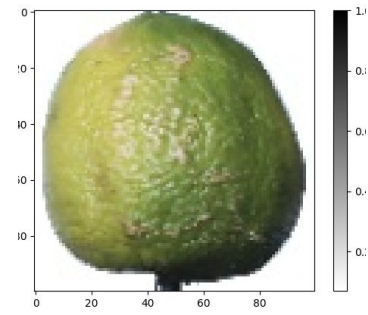
Predição 2

⇨ Esperada:
Banana
⇨ Predição:
Banana



Predição 3

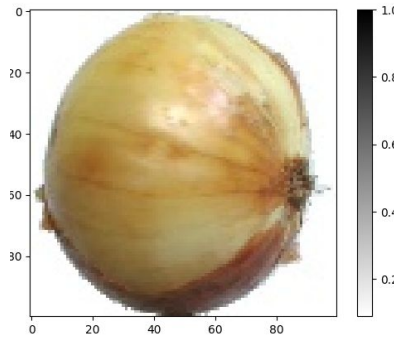
⇨ Esperada:
Peach
⇨ Predição:
Peach



Predição 4

⇨ Esperada:
Limes
⇨ Predição:
Limes

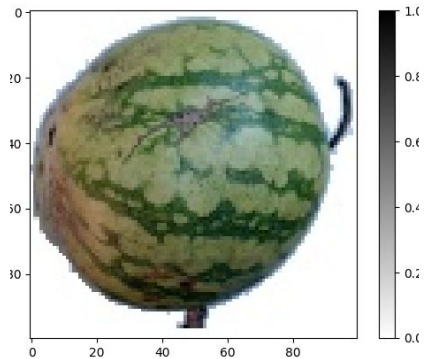
Predição: Modelo 3



Predição 1

⇨ Esperada:
Onion White

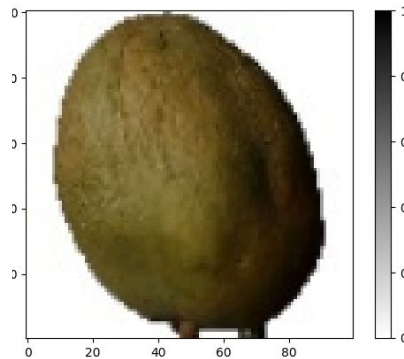
⇨ Predição:
Onion White



Predição 2

⇨ Esperada:
watermelon

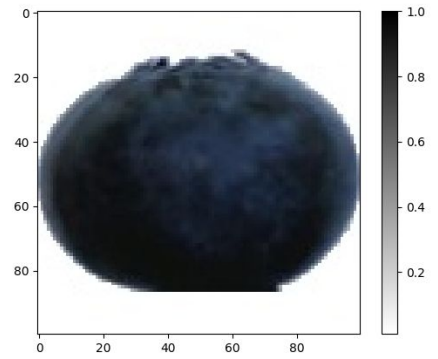
⇨ Predição:
watermelon



Predição 3

⇨ Esperada:
Papaya

⇨ Predição:
Papaya



Predição 4

⇨ Esperada:
Blueberry

⇨ Predição:
Blueberry

Conclusão

- Apesar da mudança dos parâmetros nos 3 modelos, eles continuam obtendo resultados de accuracy, F1-Score, matriz de confusão muito semelhantes.
- O método de Avaliação TP, FP, FN e TN mostrou-se o mais efetivo com o método da média geométrica, que usa os dados da avaliação anterior mostrando a diferença mais clara entre os modelos 1 que possui 0 FN , o modelo 3 que possui 3 FN e do modelo 2 que possui 54 FN o que pode impactar na precisão.
- Provável overfitting dos dados, pelo alto valor de avaliação chegando a 100%.
- Mesmo com as alterações dos parâmetros na construção dos modelos, eles obtêm resultados semelhantes , o que significa que pode haver um problema na etapa de pré-processamento.

Conclusão

- Possíveis melhorias:
 - Aumento da base de dados
 - Mudança no formato da imagens
 - Mudança no fundo, iluminação e nas cores das imagens
 - Comparação de características de classes visualmente semelhantes
- Problemas ocorridos:
 - Ambiente de execução limitado para alterações em loop de formatação de cada imagem, ou até aumento destas, o que dificulta melhorias.
 - Adequação às métricas de avaliação, pois sendo dados de imagens multiclasse , há uma limitação de opções em relação a dados tabulares ou de apenas 2 classes.
 - Dificuldade em encontrar os labels do projeto após a divisão dos dados em treino e teste, ao aplicar diferentes métricas de melhoria, avaliação e predição , por necessitar chamar as labels e imagens separadas, como por exemplo a divisão de treino_imagem, treino_labels, teste_imagem e teste_label.

Referências

- <https://mariofilho.com/precisao-recall-e-f1-score-em-machine-learning/#qual-a-f%C3%B3rmula-da-precis%C3%A3o-na-matriz-de-confus%C3%A3o>
- <https://mariofilho.com/as-metricas-mais-populares-para-avaliar-modelos-de-machine-learning/>
- <https://medium.com/data-hackers/entendendo-o-que-%C3%A9-matriz-de-confus%C3%A3o-com-python-114e683ec509>
- <https://www.kaggle.com/datasets/sshikamaru/fruit-recognition>