

Estudo comparativo sobre o desempenho de dois classificadores de frutas

Introdução

A classificação automática e precisa de frutas é uma tarefa um tanto quanto desafiadora, especialmente quando há semelhanças entre algumas variedades, como maçãs, peras e pêssegos.

Foram escolhidos dois métodos baseados em redes neurais convolucionais para reconhecimento e classificação automática de frutas para serem avaliados e em seguida um novo método baseado nos anteriores foi proposto

Dataset

Total de imagens	Tamanho do conjunto de treinamento	Tamanho do conjunto de teste	Total de classes	Tamanho das imagens	Formato de nome de arquivo de dados de treinamento	Formato de nome de arquivo de dados de teste
22495	16854 [uma fruta ou legume por imagem]	5641 [uma fruta ou legume por imagem]	33	100 x 100 pixels	[fruta/legume]_[id].jpg ex: Apple Braeburn_100.jpg	[4 digit id].jpg ex: 0001.jpg

Dataset



Carregamento e pré-processamento dos dados

Classificador

A

Classificador

B

O pré-processamento inclui redimensionamento das imagens, divisão em conjuntos de treinamento e validação, normalização, entre outros.

Arquitetura dos modelos

Classificador A

Layer (type)	Output Shape	Param #
sequential_4 (Sequential)	(None, 180, 180, 3)	0
rescaling_4 (Rescaling)	(None, 180, 180, 3)	0
conv2d_6 (Conv2D)	(None, 180, 180, 16)	448
max_pooling2d_6 (MaxPooling2D)	(None, 90, 90, 16)	0
dropout_6 (Dropout)	(None, 90, 90, 16)	0
conv2d_7 (Conv2D)	(None, 90, 90, 32)	4640
max_pooling2d_7 (MaxPooling2D)	(None, 45, 45, 32)	0
dropout_7 (Dropout)	(None, 45, 45, 32)	0
conv2d_8 (Conv2D)	(None, 45, 45, 64)	18496
max_pooling2d_8 (MaxPooling2D)	(None, 22, 22, 64)	0
dropout_8 (Dropout)	(None, 22, 22, 64)	0
flatten_2 (Flatten)	(None, 30976)	0
dense_4 (Dense)	(None, 128)	3965056
dense_5 (Dense)	(None, 33)	4257
Total params: 3,992,897		
Trainable params: 3,992,897		
Non-trainable params: 0		

Classificador B

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 100, 100, 64)	4864
conv2d_1 (Conv2D)	(None, 100, 100, 64)	102464
max_pooling2d (MaxPooling2D)	(None, 50, 50, 64)	0
dropout (Dropout)	(None, 50, 50, 64)	0
conv2d_2 (Conv2D)	(None, 50, 50, 128)	73856
conv2d_3 (Conv2D)	(None, 50, 50, 128)	147584
max_pooling2d_1 (MaxPooling2D)	(None, 25, 25, 128)	0
dropout_1 (Dropout)	(None, 25, 25, 128)	0
flatten (Flatten)	(None, 80000)	0
dense (Dense)	(None, 256)	20480256
dropout_2 (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 33)	8481
Total params: 20,817,505		
Trainable params: 20,817,505		
Non-trainable params: 0		

Compilação e treinamento do modelo

Classificador A

- Otimizador: **RMSprop**
- Função de perda: **'categorical_crossentropy'**
- 10 epochs

Classificador B

- Otimizador: **Adam**
- Função de perda: **'categorical_crossentropy'**
- Callback: **EarlyStopping**
- 12 epochs

Avaliação do desempenho do modelo

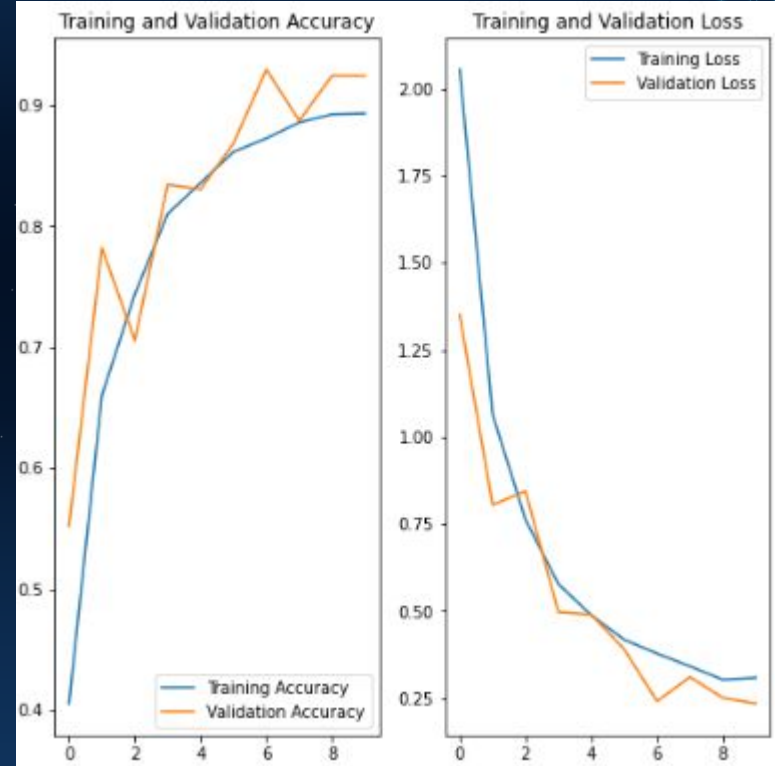
Classificador **A**

Tempo de treinamento: 26m55s

```
Epoch 1/10
422/422 [=====] - 147s 344ms/step - loss: 2.0067 - accuracy: 0.4252 - val_loss: 1.2358 - val_accuracy: 0.6451
Epoch 2/10
422/422 [=====] - 145s 343ms/step - loss: 0.9788 - accuracy: 0.6897 - val_loss: 0.8707 - val_accuracy: 0.7086
Epoch 3/10
422/422 [=====] - 145s 345ms/step - loss: 0.7118 - accuracy: 0.7629 - val_loss: 0.7533 - val_accuracy: 0.7071
Epoch 4/10
422/422 [=====] - 145s 343ms/step - loss: 0.5704 - accuracy: 0.8040 - val_loss: 0.4800 - val_accuracy: 0.8620
Epoch 5/10
422/422 [=====] - 143s 338ms/step - loss: 0.4932 - accuracy: 0.8328 - val_loss: 0.4718 - val_accuracy: 0.8306
Epoch 6/10
422/422 [=====] - 146s 345ms/step - loss: 0.4316 - accuracy: 0.8515 - val_loss: 0.3809 - val_accuracy: 0.8772
Epoch 7/10
422/422 [=====] - 143s 338ms/step - loss: 0.3766 - accuracy: 0.8756 - val_loss: 0.5133 - val_accuracy: 0.8157
Epoch 8/10
422/422 [=====] - 147s 349ms/step - loss: 0.3398 - accuracy: 0.8827 - val_loss: 0.2137 - val_accuracy: 0.9454
Epoch 9/10
422/422 [=====] - 146s 347ms/step - loss: 0.2910 - accuracy: 0.9015 - val_loss: 0.1877 - val_accuracy: 0.9531
Epoch 10/10
422/422 [=====] - 144s 342ms/step - loss: 0.2953 - accuracy: 0.8961 - val_loss: 0.1931 - val_accuracy: 0.9392
```


Avaliação do desempenho do modelo

Classificador **A**



Avaliação do desempenho do modelo

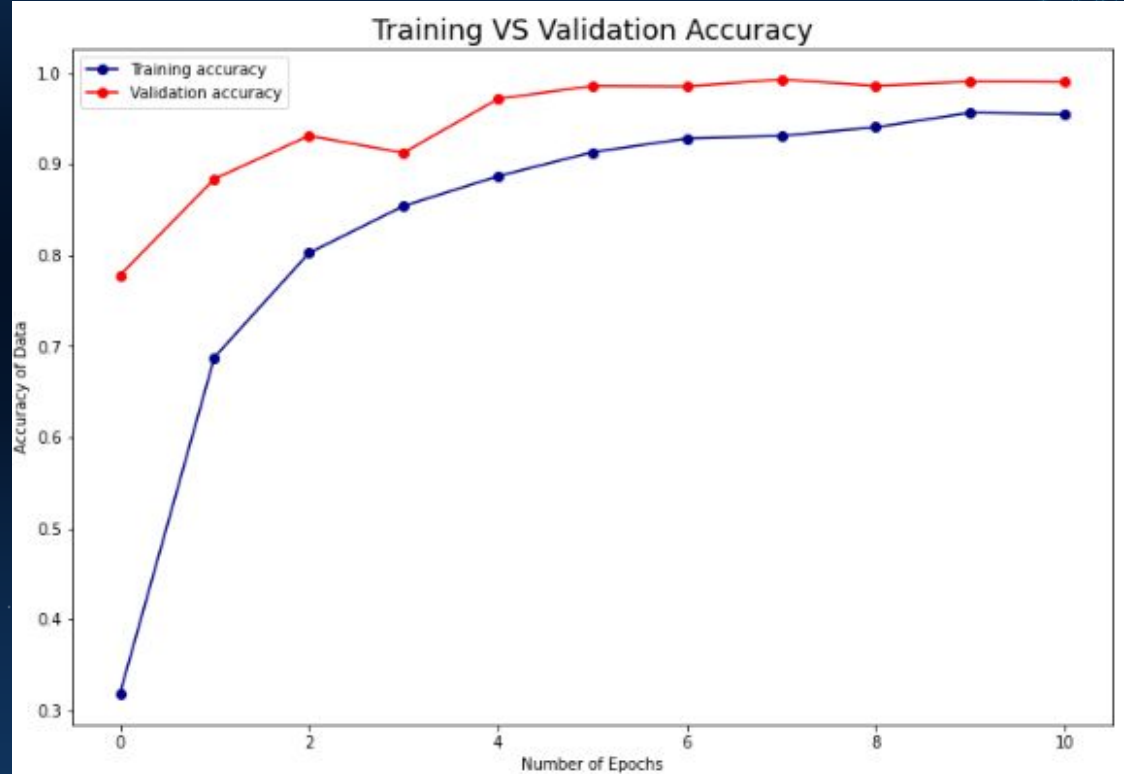
Classificador **B**

Tempo de treinamento: 4h45m

```
Epoch 1/12
421/421 [=====] - 1336s 3s/step - loss: 2.3476 - accuracy: 0.3005 - val_loss: 0.9959 - val_accuracy: 0.7351
Epoch 2/12
421/421 [=====] - 1325s 3s/step - loss: 1.0009 - accuracy: 0.6643 - val_loss: 0.3477 - val_accuracy: 0.9098
Epoch 3/12
421/421 [=====] - 1320s 3s/step - loss: 0.6040 - accuracy: 0.7936 - val_loss: 0.2098 - val_accuracy: 0.9298
Epoch 4/12
421/421 [=====] - 1315s 3s/step - loss: 0.4399 - accuracy: 0.8549 - val_loss: 0.1295 - val_accuracy: 0.9592
Epoch 5/12
421/421 [=====] - 1331s 3s/step - loss: 0.3521 - accuracy: 0.8778 - val_loss: 0.1092 - val_accuracy: 0.9673
Epoch 6/12
421/421 [=====] - 1325s 3s/step - loss: 0.2809 - accuracy: 0.9058 - val_loss: 0.0934 - val_accuracy: 0.9679
Epoch 7/12
421/421 [=====] - 1322s 3s/step - loss: 0.2357 - accuracy: 0.9198 - val_loss: 0.0551 - val_accuracy: 0.9839
Epoch 8/12
421/421 [=====] - 1327s 3s/step - loss: 0.2345 - accuracy: 0.9220 - val_loss: 0.0686 - val_accuracy: 0.9777
Epoch 9/12
421/421 [=====] - 1334s 3s/step - loss: 0.2022 - accuracy: 0.9325 - val_loss: 0.1349 - val_accuracy: 0.9449
Epoch 10/12
421/421 [=====] - 1314s 3s/step - loss: 0.1684 - accuracy: 0.9448 - val_loss: 0.0277 - val_accuracy: 0.9923
Epoch 11/12
421/421 [=====] - 1308s 3s/step - loss: 0.1578 - accuracy: 0.9458 - val_loss: 0.0639 - val_accuracy: 0.9759
Epoch 12/12
421/421 [=====] - 1302s 3s/step - loss: 0.1485 - accuracy: 0.9530 - val_loss: 0.0255 - val_accuracy: 0.9920
```

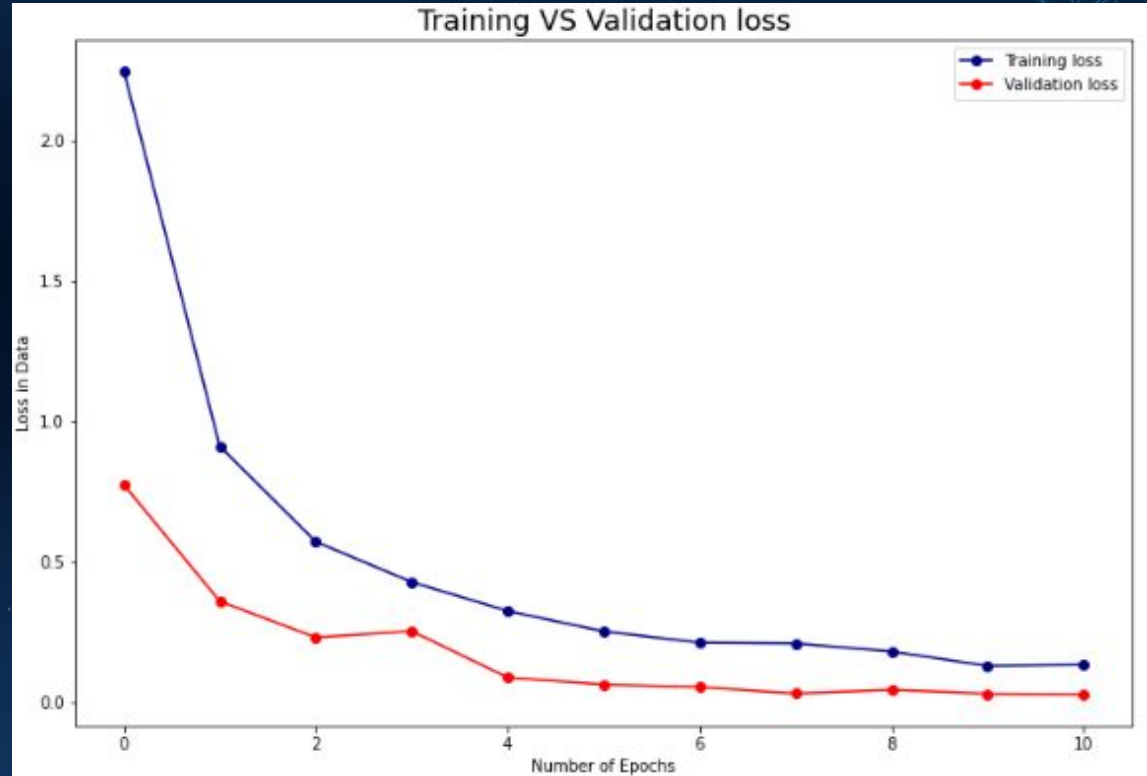
Avaliação do desempenho do modelo

Classificador **B**



Avaliação do desempenho do modelo

Classificador **B**



Previsões em imagens de teste

Classificador A



This image most likely belongs to
Pepper with a 48.29 percent
confidence



This image most likely belongs to
Grape Blue with a 98.93 percent
confidence

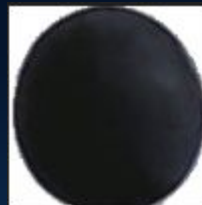


This image most likely belongs to
Raspberry with a 99.87 percent
confidence

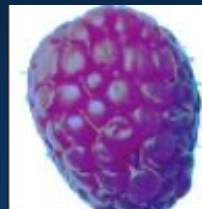
Classificador B



This image is Pepper with a
98.08 %



This image is Grape Blue with a
99.98 %



This image is Raspberry with a
100.00 %

Modificação proposta

Arquitetura

Model: "sequential_2"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 98, 98, 5)	140
max_pooling2d (MaxPooling2D)	(None, 49, 49, 5)	0
flatten (Flatten)	(None, 12005)	0
dense (Dense)	(None, 33)	396198

Total params: 396,338

Trainable params: 396,338

Non-trainable params: 0

- Otimizador: **Adam**
- Função de perda: **'categorical_crossentropy'**
- **5 épocas**

Avaliação da nova CNN

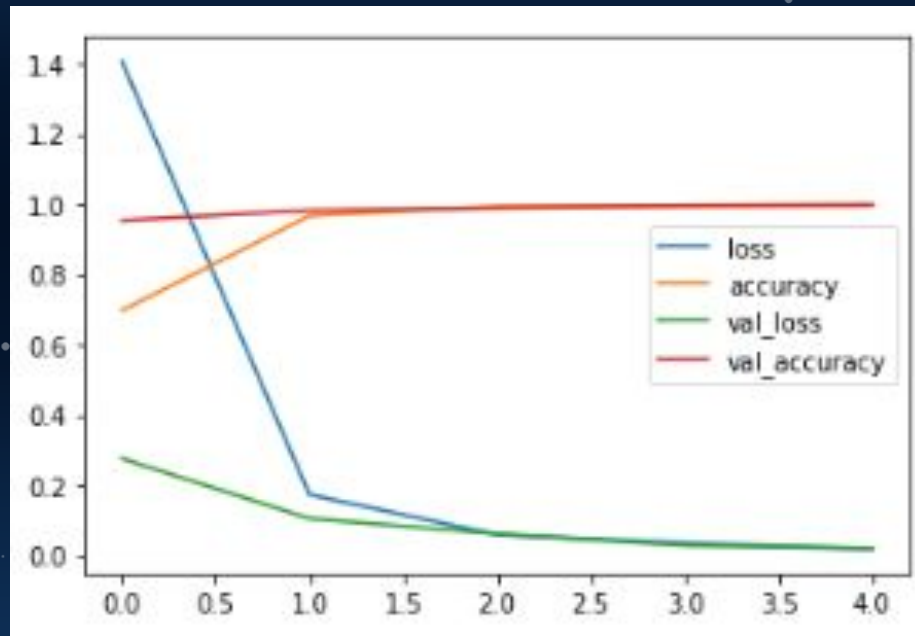
Classificador **C**

Tempo de treinamento: 51 segundos

```
Epoch 1/5
106/106 [=====] - 13s 115ms/step - loss: 1.1513 - accuracy: 0.7359 - val_loss: 0.2026 - val_accuracy: 0.9588
Epoch 2/5
106/106 [=====] - 12s 108ms/step - loss: 0.1051 - accuracy: 0.9823 - val_loss: 0.0604 - val_accuracy: 0.9935
Epoch 3/5
106/106 [=====] - 12s 114ms/step - loss: 0.0453 - accuracy: 0.9927 - val_loss: 0.0423 - val_accuracy: 0.9935
Epoch 4/5
106/106 [=====] - 12s 110ms/step - loss: 0.0230 - accuracy: 0.9973 - val_loss: 0.0175 - val_accuracy: 0.9988
Epoch 5/5
106/106 [=====] - 12s 113ms/step - loss: 0.0113 - accuracy: 0.9995 - val_loss: 0.0142 - val_accuracy: 0.9982
```

Avaliação da nova CNN

Classificador **C**



Nova CNN - Testes

Classificador **C**



Pepper 78.57 %

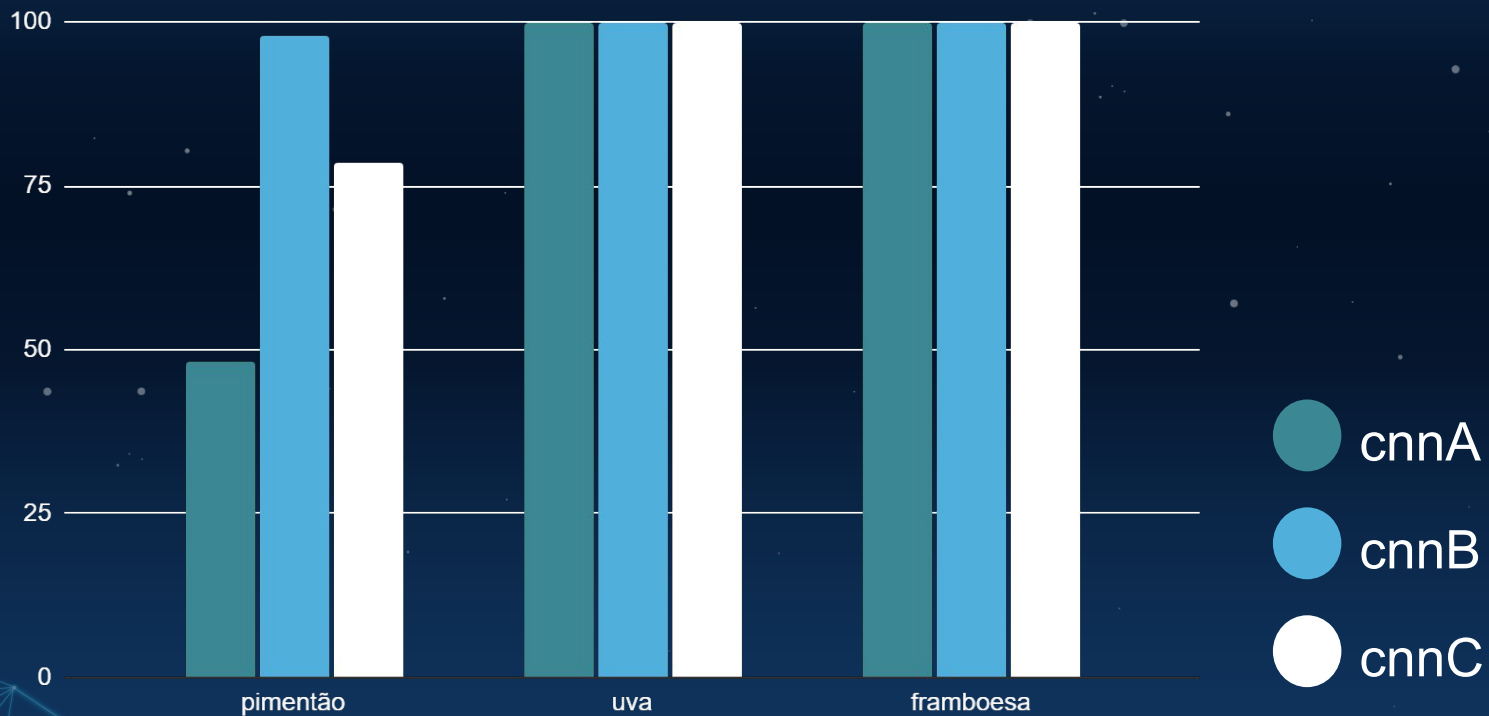


Grape Blue 99.72 %



Raspberry 99.81 %

Análise de predição



Considerações finais

- É importante encontrar um equilíbrio ao determinar o número de épocas para treinar um modelo.
- É sempre recomendável analisar métricas em conjunto com outros fatores, como tamanho do conjunto de dados, distribuição das classes, e considerar a possibilidade de ajustes adicionais no modelo ou hiperparâmetros para obter um desempenho ainda melhor.

Referências

CONVOLUTIONAL NEURAL NETWORK (CNN. Rede Neural Convolucional (CNN). Disponível em: <<https://www.tensorflow.org/tutorials/images/cnn?hl=pt-br>>. Acesso em: 19 jun. 2023.

BETTER. Melhor desempenho com a API tf.data. Disponível em: <https://www.tensorflow.org/guide/data_performance?hl=pt-br>. Acesso em: 19 jun. 2023.

seaborn: statistical data visualization — seaborn 0.12.2 documentation. Disponível em: <<https://seaborn.pydata.org/>>. Acesso em: 19 jun. 2023.

scikit-learn: machine learning in Python — scikit-learn 1.2.2 documentation. Disponível em: <<https://scikit-learn.org/stable/>>. Acesso em: 19 jun. 2023.

Pillow. Disponível em: <<https://pillow.readthedocs.io/en/stable/>>. Acesso em: 19 jun. 2023.