Com Fine Tunning Sem Data Augmentation

Neste modelo ao implementarmos o fine tunning descongelámos apenas a primeira camada, e fizemos uso da camada classificadora do modelo sem fine Tunning.

Carregar as imagens

```
from tensorflow import keras

from keras.applications.vgg19 import VGG19
from keras.optimizers import RMSprop
from keras.utils import image_dataset_from_directory
```

Batch Size

A decisão de um *batch size* a 32 foi devido às experiências e treinos anteriores

```
IMG_SIZE = 150
num_classes = 10
# Carregar e preparar os dados

train_dir = '../Imagens/train/train5'
validation_dir = '../Imagens/validation'
test_dir = '../Imagens/test'

train_dataset = image_dataset_from_directory(train_dir, image_size=(IMG_SIZE, IMG_SIZE), batch_size=32,label_mode='categorical')
validation_dataset = image_dataset_from_directory(validation_dir, image_size=(IMG_SIZE, IMG_SIZE), batch_size=32,label_mode='categorical')
test_dataset = image_dataset_from_directory(test_dir, image_size=(IMG_SIZE, IMG_SIZE), batch_size=32,label_mode='categorical')
```

```
Found 40000 files belonging to 10 classes.
Found 10000 files belonging to 10 classes.
Found 10000 files belonging to 10 classes.
```

Figura 1 - Carregar as imagens

Carregar a camada classificadora já treinada.

```
from tensorflow import keras
from keras import layers
from keras import layers, regularizers
from keras.callbacks import ReduceLROnPlateau, EarlyStopping, ModelCheckpoint

model = keras.models.load_model('TL_dataAugmentation.h5')
```

Figura 2 - Carregar a classificadora

Nas imagens seguintes podemos verificar que apenas a última camada vai ser treinável.

```python
convbase = model.get_layer("vgg19")

for layer in convbase.layers:
    if layer.name in ['block5_conv4']:
        layer.trainable = True
    else:
        layer.trainable = False

for i, layer in enumerate(convbase.layers):
    print(i, layer.name, layer.trainable)
```

Figura 4 - Descongelar a última camada

```
0 input_24 False
1 block1_conv1 False
2 block1_conv2 False
3 block1_pool False
4 block2_conv1 False
5 block2_conv2 False
6 block2_pool False
7 block3_conv1 False
8 block3_conv2 False
9 block3_conv3 False
10 block3_conv4 False
11 block3_pool False
12 block4_conv1 False
13 block4_conv2 False
14 block4_conv3 False
15 block4_conv4 False
16 block4_pool False
17 block5_conv1 False
18 block5_conv2 False
19 block5_conv3 False
20 block5_conv4 True
21 block5_pool False
```

Figura 3 - Layout da Vgg19

A escolha do otimizador foi derivada do último modelo, onde obtivemos melhores resultados com o RMSprop.

```python
from keras.utils import to_categorical
from tensorflow import keras
from keras import optimizers
from keras.optimizers import Adam


# Specify the learning rate
learning_rate = 0.000001

# Define the optimizer with the specified learning rate
optimizer = keras.optimizers.RMSprop(learning_rate=learning_rate)

# Compile the model with the optimizer and learning rate
model.compile(optimizer=optimizer,
              loss='categorical_crossentropy',
              metrics=['accuracy'])
history = model.fit(train_dataset, epochs=30, validation_data=validation_dataset)# mais epocas
model.save('Sem_dataAugmentation.h5')
```

Figura 5 - Alguns hiperparâmetros

Output

```
val_loss, val_acc = model.evaluate(validation_dataset)
print('val_acc:'
, val_acc)
```

```
313/313 [==============================] - 22s 69ms/step - loss: 0.3042 - accuracy: 0.9254
val_acc: 0.9254000186920166
```

Figura 6 - Validação

```
Epoch 1/30
WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.
1250/1250 [==============================] - 222s 176ms/step - loss: 0.1990 - accuracy: 0.9430 - val_loss: 0.2768 - val_accuracy: 0.9213
Epoch 2/30
1250/1250 [==============================] - 218s 174ms/step - loss: 0.1888 - accuracy: 0.9464 - val_loss: 0.2762 - val_accuracy: 0.9224
Epoch 3/30
...
Epoch 29/30
1250/1250 [==============================] - 216s 173ms/step - loss: 0.1456 - accuracy: 0.9617 - val_loss: 0.3082 - val_accuracy: 0.9247
Epoch 30/30
1250/1250 [==============================] - 217s 173ms/step - loss: 0.1492 - accuracy: 0.9606 - val_loss: 0.3042 - val_accuracy: 0.9254
```

Figura 8 - Output

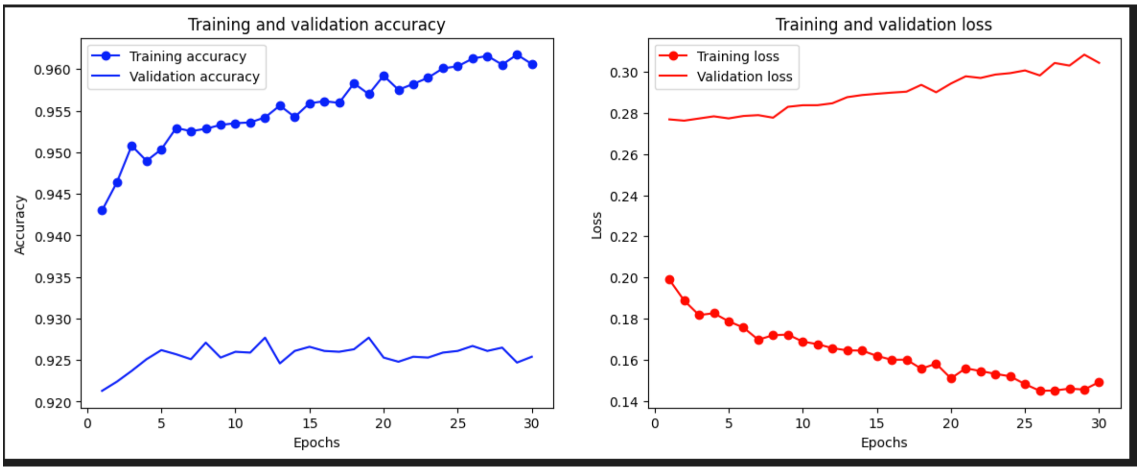

Figura 7 – Gráfico

Treino 1

A conclusão a que chegámos ao longo dos treinos é que para uma rede com fine tunning conseguimos ajustar bem mais a VGG19 ao nosso problema das 10 labels, obtendo assim melhores resultados.

Neste modelo a única coisa que muda do modelo final é o learning rate, e reparámos que com este lr menor houve uma demora maior na convergência, mas não houve tanto overfitting como no modelo final.

```
from keras.utils import to_categorical
from tensorflow import keras
from keras import optimizers
from keras.optimizers import Adam


# Specify the learning rate
learning_rate = 0.000001

# Define the optimizer with the specified learning rate
optimizer = keras.optimizers.RMSprop(learning_rate=learning_rate)

# Compile the model with the optimizer and learning rate
model.compile(optimizer=optimizer,
              loss='categorical_crossentropy',
              metrics=['accuracy'])
history = model.fit(train_dataset, epochs=30, validation_data=validation_dataset)# mais epocas
model.save('Sem_dataAugmentation.h5')
```

Figura 9 - Alguns Hiperparâmetros

```
val_loss, val_acc = model.evaluate(validation_dataset)
print('val_acc:'
, val_acc)
```

```
313/313 [==============================] - 23s 72ms/step - loss: 0.3553 - accuracy: 0.8903
val_acc: 0.8902999758720398
```

Figura 10 - Output

Gráfico

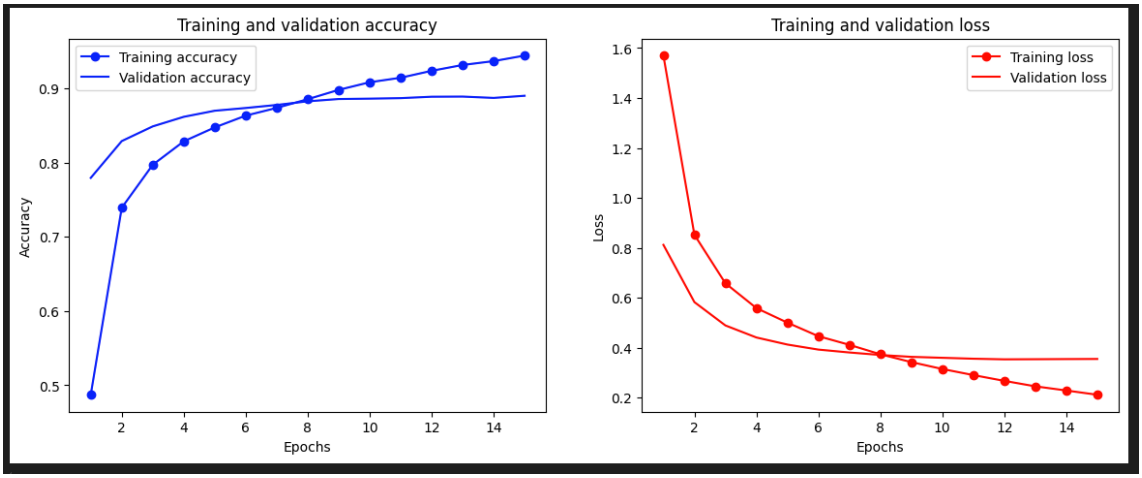

Figura 11 - Gráfico Treino 1

Treino 2

No segundo treino, aumentei o learning rate o que ajudou bastante a atingir melhores resultados e aumentou a velocidade a que a rede convergiu.

```
from keras.utils import to_categorical
from tensorflow import keras
from keras import optimizers
from keras.optimizers import Adam


# Specify the learning rate
learning_rate = 0.00001

# Define the optimizer with the specified learning rate
optimizer = keras.optimizers.RMSprop(learning_rate=learning_rate)

# Compile the model with the optimizer and learning rate
model.compile(optimizer=optimizer,
              loss='categorical_crossentropy',
              metrics=['accuracy'])
history = model.fit(train_dataset, epochs=30, validation_data=validation_dataset)
model.save('Sem_dataAugmentation.h5')
```

Figura 12 - Hiperparâmetros treino 2

```
Epoch 1/30
WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.
1250/1250 [==============================] - 222s 176ms/step - loss: 0.1990 - accuracy: 0.9430 - val_loss: 0.2768 - val_accuracy: 0.9213
Epoch 2/30
1250/1250 [==============================] - 218s 174ms/step - loss: 0.1888 - accuracy: 0.9464 - val_loss: 0.2762 - val_accuracy: 0.9224
Epoch 3/30
...
Epoch 29/30
1250/1250 [==============================] - 216s 173ms/step - loss: 0.1456 - accuracy: 0.9617 - val_loss: 0.3082 - val_accuracy: 0.9247
Epoch 30/30
1250/1250 [==============================] - 217s 173ms/step - loss: 0.1492 - accuracy: 0.9606 - val_loss: 0.3042 - val_accuracy: 0.9254
```

Figura 13 - Output do treino 2

```
val_loss, val_acc = model.evaluate(validation_dataset)
print('val_acc:'
, val_acc)
```

```
313/313 [==============================] - 22s 69ms/step - loss: 0.3042 - accuracy: 0.9254
val_acc: 0.9254000186920166
```

Figura 14 - Validação treino 2

Gráfico

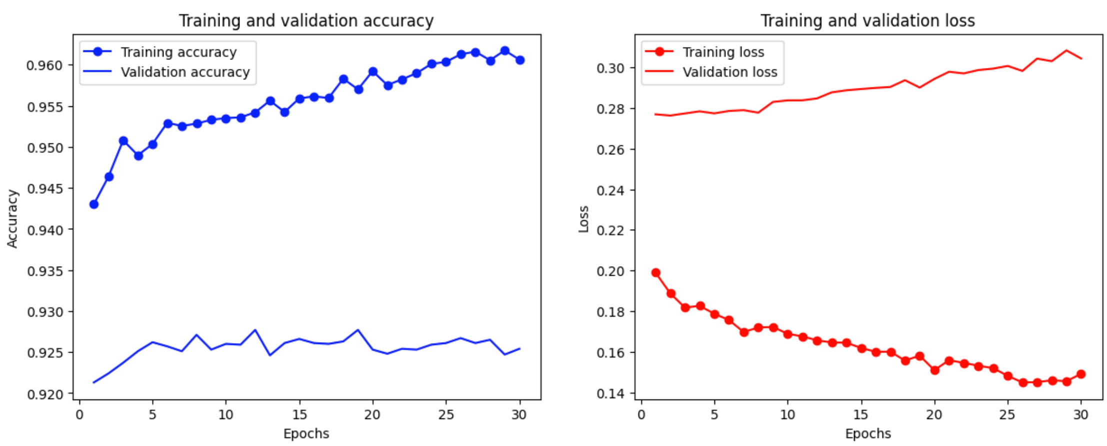

Figura 15 - Gráfico treino 2

Treino 3

Neste treino reparámos em algo bastante interessante porque tentámos descongelar mais duas camadas e o resultado obtido não foi o melhor.

```
convbase = model.get_layer("vgg19")

for layer in convbase.layers:
    if layer.name in ['block5_conv4', 'block5_conv3', 'block5_conv2', 'block5_conv1']:
        layer.trainable = True
    else:
        layer.trainable = False

for i, layer in enumerate(convbase.layers):
    print(i, layer.name, layer.trainable)
```

Figura 16 - Descongelar as camadas da VGG19

Aumentámos também o learing rate o que ajudou ainda mais para o mau resultado obtido já que com um learing rate muito alto o modelo pode ajustar-se rapidamente aos dados de treino, mas não faça com que generalize muito bem os dados.

```
from keras.utils import to_categorical
from tensorflow import keras
from keras import optimizers
from keras.optimizers import Adam
import tensorflow as tf

loaded_model = tf.keras.models.load_model('Sem_dataAugmentation.h5')

# Specify the learning rate
learning_rate = 0.0001

# Define the optimizer with the specified learning rate
optimizer = keras.optimizers.RMSprop(learning_rate=learning_rate)

# Compile the model with the optimizer and learning rate
model.compile(optimizer=optimizer,
              loss='categorical_crossentropy',
              metrics=['accuracy'])
history = model.fit(train_dataset, epochs=30, validation_data=validation_dataset)# mais epocas
model.save('Sem_dataAugmentation.h5')
```

Figura 17 - Alguns hiperparâmetros do treino 3

```
Epoch 1/30
WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.
1250/1250 [==============================] - 247s 196ms/step - loss: 0.4605 - accuracy: 0.8564 - val_loss: 0.5786 - val_accuracy: 0.8377
Epoch 2/30
1250/1250 [==============================] - 240s 192ms/step - loss: 0.4271 - accuracy: 0.8673 - val_loss: 0.4569 - val_accuracy: 0.8676
Epoch 3/30
...
1250/1250 [==============================] - 243s 194ms/step - loss: 0.2811 - accuracy: 0.9151 - val_loss: 0.5055 - val_accuracy: 0.8693
Epoch 11/30
1250/1250 [==============================] - 230s 184ms/step - loss: 0.2753 - accuracy: 0.9185 - val_loss: 0.3658 - val_accuracy: 0.8987
```

Figura 18 – Output

Na 11ª época decidimos parar a execução porque notámos que a rede não estava a preformar conforme o esperado já que a aprendizagem entre épocas não estava a fluir como devia.