

```
In [1]: import tensorflow as tf
from tensorflow.keras import regularizers
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow import keras
from keras import layers
from keras import models
from tensorflow.keras import optimizers
from sklearn.metrics import classification_report
import numpy as np
```

2025-06-12 00:47:09.855291: E external/local_xla/xla/stream_executor/cuda/cuda_fft.cc:467] Unable to register cuFFT factory: Attempting to register factory for plugin cuFFT when one has already been registered
 WARNING: All log messages before absl::InitializeLog() is called are written to STDERR
 E0000 00:00:1749685629.868570 47878 cuda_dnn.cc:8579] Unable to register cuDNN factory: Attempting to register factory for plugin cuDNN when one has already been registered
 E0000 00:00:1749685629.872765 47878 cuda_blas.cc:1407] Unable to register cuBLAS factory: Attempting to register factory for plugin cuBLAS when one has already been registered
 W0000 00:00:1749685629.882646 47878 computation_placer.cc:177] computation placer already registered. Please check linkage and avoid linking the same target more than once.
 W0000 00:00:1749685629.882661 47878 computation_placer.cc:177] computation placer already registered. Please check linkage and avoid linking the same target more than once.
 W0000 00:00:1749685629.882662 47878 computation_placer.cc:177] computation placer already registered. Please check linkage and avoid linking the same target more than once.
 W0000 00:00:1749685629.882663 47878 computation_placer.cc:177] computation placer already registered. Please check linkage and avoid linking the same target more than once.
 2025-06-12 00:47:09.886214: I tensorflow/core/platform/cpu_feature_guard.cc:210] This TensorFlow binary is optimized to use available CPU instructions in performance-critical operations.
 To enable the following instructions: AVX2 FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.

```
In [2]: tf.__version__
```

Out[2]: '2.19.0'

```
In [3]: train_dir = 'dataset_balanceado_final/train'
validation_dir = 'dataset_balanceado_final/validation'
test_dir = 'dataset_balanceado_final/test'
```

Definir batch_size e image_size

```
In [4]: from tensorflow.keras.utils import image_dataset_from_directory

IMG_SIZE = 150
BATCH_SIZE = 32
```

Training set - É o conjunto de dados usado para treinar a rede

```
In [5]: train_dataset = image_dataset_from_directory(
    train_dir,
    image_size=(IMG_SIZE, IMG_SIZE),
    batch_size=BATCH_SIZE,
    label_mode='categorical'
)
```

Found 4276 files belonging to 7 classes.

```
I0000 00:00:1749685632.292233 47878 gpu_device.cc:2019] Created device /job:localhost/replica:0/task:0/device:GPU:0 with 4804 MB memory: -> device: 0, name: NVidia GeForce GTX 1660 SUPER, pci bus id: 0000:03:00.0, compute capability: 7.5
I0000 00:00:1749685632.295785 47878 gpu_device.cc:2019] Created device /job:localhost/replica:0/task:0/device:GPU:1 with 4804 MB memory: -> device: 1, name: NVidia GeForce GTX 1660 SUPER, pci bus id: 0000:05:00.0, compute capability: 7.5
```

Validation set - Usado para 'testar' o modelo durante o processo de procura da melhor combinação de hiperparâmetros.

```
In [6]: validation_dataset = image_dataset_from_directory(
    validation_dir,
    image_size=(IMG_SIZE, IMG_SIZE),
    batch_size=BATCH_SIZE,
    label_mode='categorical'
)
```

Found 1420 files belonging to 7 classes.

Test set - Usado para testar o modelo depois do processo de treino

```
In [7]: test_dataset = image_dataset_from_directory(
    test_dir,
    image_size=(IMG_SIZE, IMG_SIZE),
    batch_size=BATCH_SIZE,
    label_mode='categorical'
)
```

Found 1420 files belonging to 7 classes.

Métricas para avaliar os modelos

```
In [8]: # Utiliza uma função (do scikit-Learn) para avaliar o desempenho do modelo, indicando
# f1-score do modelo
# accuracy do modelo
# accuracy por classe

from sklearn.metrics import classification_report
import numpy as np

def print_classification_metrics(model, dataset, phase_name):
    y_true = []
    y_pred = []

    for images, labels in dataset:
        preds = model.predict(images)
        y_true.extend(np.argmax(labels.numpy(), axis=1))
        y_pred.extend(np.argmax(preds, axis=1))
```

```
print(f"\n {phase_name}")
print(classification_report(y_true, y_pred, digits=4))
```

MODELO 2 (VGG16) - Optuna (sem data augmentation)

Feature Extraction

```
In [9]: from tensorflow import keras
from keras import layers
from keras.applications import VGG16 # Importa a arquitetura VGG16 pré-treinada

# Carregar a base VGG16 pré-treinada
conv_base = VGG16(weights='imagenet', include_top=False, input_shape=(150, 150,
conv_base.trainable = False # Congela todas as camadas da VGG16, impedindo que o

# Usar data augmentation
inputs = layers.Input(shape=(150, 150, 3)) # Define a camada de entrada do novo
x = keras.applications.vgg16.preprocess_input(inputs) # Aplica o pré-processamen
x = conv_base(x) # Passa as imagens (pré-processadas e aumentadas) através da ba
x = layers.Flatten()(x) # Achata as características extraídas para um vetor 1D.
x = layers.Dense(512, activation='relu')(x) # Adiciona uma camada densa com 512
x = layers.Dropout(0.5)(x) # Aplica Dropout (50%) para regularização e prevenção
outputs = layers.Dense(7, activation='softmax')(x) # Adiciona a camada de saída

model_t = models.Model(inputs, outputs) # Cria o model_t

# Compilar e treinar (feature extraction)
model_t.compile( # Compila o modelo para configurar o processo de treino.
    loss='categorical_crossentropy', # Define a função de perda apropriada para
    optimizer=keras.optimizers.RMSprop(learning_rate=0.000934755), # Configura o
    metrics=['accuracy'] # Define 'accuracy' (precisão) como a métrica a ser usa
)

history_t = model_t.fit( # Treina o modelo.
    train_dataset, # Usa o conjunto de dados de treino.
    validation_data=validation_dataset, # Usa o conjunto de dados de validação p
    epochs=8 # Treina o modelo por 10 épocas.
)
```

Epoch 1/8

WARNING: All log messages before absl::InitializeLog() is called are written to STDERR

I0000 00:00:1749685635.010232 48055 service.cc:152] XLA service 0x73e1900046d0 initialized for platform CUDA (this does not guarantee that XLA will be used). Devices:

I0000 00:00:1749685635.010251 48055 service.cc:160] StreamExecutor device (0): NVIDIA GeForce GTX 1660 SUPER, Compute Capability 7.5

I0000 00:00:1749685635.010254 48055 service.cc:160] StreamExecutor device (1): NVIDIA GeForce GTX 1660 SUPER, Compute Capability 7.5

2025-06-12 00:47:15.053971: I tensorflow/compiler/mlir/tensorflow/utils/dump_mlir_util.cc:269] disabling MLIR crash reproducer, set env var `MLIR_CRASH_REPRODUCER_DIRECTORY` to enable.

I0000 00:00:1749685635.296612 48055 cuda_dnn.cc:529] Loaded cuDNN version 90300

2025-06-12 00:47:15.590619: I external/local_xla/xla/service/gpu/autotuning/conv_algorithm_picker.cc:549] Omitted potentially buggy algorithm eng14{k25=0} for conv %cudnn-conv-bias-activation.39 = (f32[32,64,150,150]{3,2,1,0}, u8[0]{0}) custom-call(f32[32,3,150,150]{3,2,1,0} %bitcast.3516, f32[64,3,3,3]{3,2,1,0} %bitcast.3523, f32[64]{0} %bitcast.3525), window={size=3x3 pad=1_1x1_1}, dim_labels=bf01_o101->bf01, custom_call_target="__cudnn\$convBiasActivationForward", metadata={op_type="Conv2D" op_name="functional_1/vgg16_1/block1_conv1_1/convolution" source_file="/home/eliana/escola/venv/lib/python3.12/site-packages/tensorflow/python/framework/ops.py" source_line=1200}, backend_config={"operation_queue_id": "0", "wait_on_operation_queues": [], "cudnn_conv_backend_config": {"conv_result_scale": 1, "activation_mode": "kRelu", "side_input_scale": 0, "leakyrelu_alpha": 0}, "force_earliest_schedule": false}

2025-06-12 00:47:15.687945: I external/local_xla/xla/service/gpu/autotuning/conv_algorithm_picker.cc:549] Omitted potentially buggy algorithm eng14{k25=0} for conv %cudnn-conv-bias-activation.40 = (f32[32,64,150,150]{3,2,1,0}, u8[0]{0}) custom-call(f32[32,64,150,150]{3,2,1,0} %bitcast.3530, f32[64,64,3,3]{3,2,1,0} %bitcast.3537, f32[64]{0} %bitcast.3539), window={size=3x3 pad=1_1x1_1}, dim_labels=bf01_o101->bf01, custom_call_target="__cudnn\$convBiasActivationForward", metadata={op_type="Conv2D" op_name="functional_1/vgg16_1/block1_conv2_1/convolution" source_file="/home/eliana/escola/venv/lib/python3.12/site-packages/tensorflow/python/framework/ops.py" source_line=1200}, backend_config={"operation_queue_id": "0", "wait_on_operation_queues": [], "cudnn_conv_backend_config": {"conv_result_scale": 1, "activation_mode": "kRelu", "side_input_scale": 0, "leakyrelu_alpha": 0}, "force_earliest_schedule": false}

2025-06-12 00:47:16.601457: I external/local_xla/xla/service/gpu/autotuning/conv_algorithm_picker.cc:549] Omitted potentially buggy algorithm eng14{k25=0} for conv %cudnn-conv-bias-activation.41 = (f32[32,128,75,75]{3,2,1,0}, u8[0]{0}) custom-call(f32[32,64,75,75]{3,2,1,0} %bitcast.3547, f32[128,64,3,3]{3,2,1,0} %bitcast.3554, f32[128]{0} %bitcast.3556), window={size=3x3 pad=1_1x1_1}, dim_labels=bf01_o101->bf01, custom_call_target="__cudnn\$convBiasActivationForward", metadata={op_type="Conv2D" op_name="functional_1/vgg16_1/block2_conv1_1/convolution" source_file="/home/eliana/escola/venv/lib/python3.12/site-packages/tensorflow/python/framework/ops.py" source_line=1200}, backend_config={"operation_queue_id": "0", "wait_on_operation_queues": [], "cudnn_conv_backend_config": {"conv_result_scale": 1, "activation_mode": "kRelu", "side_input_scale": 0, "leakyrelu_alpha": 0}, "force_earliest_schedule": false}

2025-06-12 00:47:17.005866: I external/local_xla/xla/service/gpu/autotuning/conv_algorithm_picker.cc:549] Omitted potentially buggy algorithm eng14{k25=0} for conv %cudnn-conv-bias-activation.42 = (f32[32,128,75,75]{3,2,1,0}, u8[0]{0}) custom-call(f32[32,128,75,75]{3,2,1,0} %bitcast.3561, f32[128,128,3,3]{3,2,1,0} %bitcast.3568, f32[128]{0} %bitcast.3570), window={size=3x3 pad=1_1x1_1}, dim_labels=bf01_o101->bf01, custom_call_target="__cudnn\$convBiasActivationForward", metadata={op_type="Conv2D" op_name="functional_1/vgg16_1/block2_conv2_1/convolution" source_file="/home/eliana/escola/venv/lib/python3.12/site-packages/tensorflow/python/framework/ops.py" source_line=1200}, backend_config={"operation_queue_id": "0", "wait_on_operation_queues": [], "cudnn_conv_backend_config": {"conv_result_scale": 1, "activation_mode": "kRelu", "side_input_scale": 0, "leakyrelu_alpha": 0}, "force_earliest_schedule": false}

```

edule":false}
2025-06-12 00:47:17.731223: I external/local_xla/xla/service/gpu/autotuning/conv_
algorithm_picker.cc:549] Omitted potentially buggy algorithm eng14{k25=0} for con
v %cudnn-conv-bias-activation.43 = (f32[32,256,37,37]{3,2,1,0}, u8[0]{0}) custom-
call(f32[32,128,37,37]{3,2,1,0} %bitcast.3576, f32[256,128,3,3]{3,2,1,0} %bitcas
t.3583, f32[256]{0} %bitcast.3585), window={size=3x3 pad=1_1x1_1}, dim_labels=bf0
1_o101->bf01, custom_call_target="__cudnn$convBiasActivationForward", metadata={o
p_type="Conv2D" op_name="functional_1/vgg16_1/block3_conv1_1/convolution" source_
file="/home/eliana/escola/venv/lib/python3.12/site-packages/tensorflow/python/fra
mework/ops.py" source_line=1200}, backend_config={"operation_queue_id":"0","wait_
on_operation_queues":[],"cudnn_conv_backend_config":{"conv_result_scale":1,"activ
ation_mode":"kRelu","side_input_scale":0,"leakyrelu_alpha":0},"force_earliest_sch
edule":false}
2025-06-12 00:47:18.062808: I external/local_xla/xla/service/gpu/autotuning/conv_
algorithm_picker.cc:549] Omitted potentially buggy algorithm eng14{k25=0} for con
v %cudnn-conv-bias-activation.44 = (f32[32,256,37,37]{3,2,1,0}, u8[0]{0}) custom-
call(f32[32,256,37,37]{3,2,1,0} %bitcast.3590, f32[256,256,3,3]{3,2,1,0} %bitcas
t.3597, f32[256]{0} %bitcast.3599), window={size=3x3 pad=1_1x1_1}, dim_labels=bf0
1_o101->bf01, custom_call_target="__cudnn$convBiasActivationForward", metadata={o
p_type="Conv2D" op_name="functional_1/vgg16_1/block3_conv2_1/convolution" source_
file="/home/eliana/escola/venv/lib/python3.12/site-packages/tensorflow/python/fra
mework/ops.py" source_line=1200}, backend_config={"operation_queue_id":"0","wait_
on_operation_queues":[],"cudnn_conv_backend_config":{"conv_result_scale":1,"activ
ation_mode":"kRelu","side_input_scale":0,"leakyrelu_alpha":0},"force_earliest_sch
edule":false}
2025-06-12 00:47:18.591254: I external/local_xla/xla/service/gpu/autotuning/conv_
algorithm_picker.cc:549] Omitted potentially buggy algorithm eng14{k25=0} for con
v %cudnn-conv-bias-activation.46 = (f32[32,512,18,18]{3,2,1,0}, u8[0]{0}) custom-
call(f32[32,256,18,18]{3,2,1,0} %bitcast.3619, f32[512,256,3,3]{3,2,1,0} %bitcas
t.3626, f32[512]{0} %bitcast.3628), window={size=3x3 pad=1_1x1_1}, dim_labels=bf0
1_o101->bf01, custom_call_target="__cudnn$convBiasActivationForward", metadata={o
p_type="Conv2D" op_name="functional_1/vgg16_1/block4_conv1_1/convolution" source_
file="/home/eliana/escola/venv/lib/python3.12/site-packages/tensorflow/python/fra
mework/ops.py" source_line=1200}, backend_config={"operation_queue_id":"0","wait_
on_operation_queues":[],"cudnn_conv_backend_config":{"conv_result_scale":1,"activ
ation_mode":"kRelu","side_input_scale":0,"leakyrelu_alpha":0},"force_earliest_sch
edule":false}
2025-06-12 00:47:18.899377: I external/local_xla/xla/service/gpu/autotuning/conv_
algorithm_picker.cc:549] Omitted potentially buggy algorithm eng14{k25=0} for con
v %cudnn-conv-bias-activation.47 = (f32[32,512,18,18]{3,2,1,0}, u8[0]{0}) custom-
call(f32[32,512,18,18]{3,2,1,0} %bitcast.3633, f32[512,512,3,3]{3,2,1,0} %bitcas
t.3640, f32[512]{0} %bitcast.3642), window={size=3x3 pad=1_1x1_1}, dim_labels=bf0
1_o101->bf01, custom_call_target="__cudnn$convBiasActivationForward", metadata={o
p_type="Conv2D" op_name="functional_1/vgg16_1/block4_conv2_1/convolution" source_
file="/home/eliana/escola/venv/lib/python3.12/site-packages/tensorflow/python/fra
mework/ops.py" source_line=1200}, backend_config={"operation_queue_id":"0","wait_
on_operation_queues":[],"cudnn_conv_backend_config":{"conv_result_scale":1,"activ
ation_mode":"kRelu","side_input_scale":0,"leakyrelu_alpha":0},"force_earliest_sch
edule":false}
2025-06-12 00:47:19.329176: I external/local_xla/xla/service/gpu/autotuning/conv_
algorithm_picker.cc:549] Omitted potentially buggy algorithm eng14{k25=0} for con
v %cudnn-conv-bias-activation.49 = (f32[32,512,9,9]{3,2,1,0}, u8[0]{0}) custom-ca
ll(f32[32,512,9,9]{3,2,1,0} %bitcast.3662, f32[512,512,3,3]{3,2,1,0} %bitcast.366
9, f32[512]{0} %bitcast.3671), window={size=3x3 pad=1_1x1_1}, dim_labels=bf01_o10
1->bf01, custom_call_target="__cudnn$convBiasActivationForward", metadata={op_typ
e="Conv2D" op_name="functional_1/vgg16_1/block5_conv1_1/convolution" source_file
="/home/eliana/escola/venv/lib/python3.12/site-packages/tensorflow/python/fra
mework/ops.py" source_line=1200}, backend_config={"operation_queue_id":"0","wait_on_o
peration_queues":[],"cudnn_conv_backend_config":{"conv_result_scale":1,"activatio

```

```
n_mode": "kRelu", "side_input_scale": 0, "leakyrelu_alpha": 0}, "force_earliest_schedule": false}  
2/134 ————— 14s 108ms/step - accuracy: 0.1328 - loss: 28.3495  
I0000 00:00:1749685640.414680 48055 device_compiler.h:188] Compiled cluster using XLA! This line is logged at most once for the lifetime of the process.  
133/134 ————— 0s 106ms/step - accuracy: 0.3727 - loss: 13.0227
```

2025-06-12 00:47:34.938689: I external/local_xla/xla/service/gpu/autotuning/conv_algorithm_picker.cc:549] Omitted potentially buggy algorithm eng14{k25=0} for conv %cudnn-conv-bias-activation.39 = (f32[20,64,150,150]{3,2,1,0}, u8[0]{0}) custom-call(f32[20,3,150,150]{3,2,1,0} %bitcast.3516, f32[64,3,3,3]{3,2,1,0} %bitcast.3523, f32[64]{0} %bitcast.3525), window={size=3x3 pad=1_1x1_1}, dim_labels=bf01_o1_o1->bf01, custom_call_target="__cudnn\$convBiasActivationForward", metadata={op_type="Conv2D" op_name="functional_1/vgg16_1/block1_conv1_1/convolution" source_file="/home/eliana/escola/venv/lib/python3.12/site-packages/tensorflow/python/framework/ops.py" source_line=1200}, backend_config={"operation_queue_id": "0", "wait_on_operation_queues": [], "cudnn_conv_backend_config": {"conv_result_scale": 1, "activation_mode": "kRelu", "side_input_scale": 0, "leakyrelu_alpha": 0}, "force_earliest_schedule": false}

2025-06-12 00:47:35.039389: I external/local_xla/xla/service/gpu/autotuning/conv_algorithm_picker.cc:549] Omitted potentially buggy algorithm eng14{k25=0} for conv %cudnn-conv-bias-activation.40 = (f32[20,64,150,150]{3,2,1,0}, u8[0]{0}) custom-call(f32[20,64,150,150]{3,2,1,0} %bitcast.3530, f32[64,64,3,3]{3,2,1,0} %bitcast.3537, f32[64]{0} %bitcast.3539), window={size=3x3 pad=1_1x1_1}, dim_labels=bf01_o1_o1->bf01, custom_call_target="__cudnn\$convBiasActivationForward", metadata={op_type="Conv2D" op_name="functional_1/vgg16_1/block1_conv2_1/convolution" source_file="/home/eliana/escola/venv/lib/python3.12/site-packages/tensorflow/python/framework/ops.py" source_line=1200}, backend_config={"operation_queue_id": "0", "wait_on_operation_queues": [], "cudnn_conv_backend_config": {"conv_result_scale": 1, "activation_mode": "kRelu", "side_input_scale": 0, "leakyrelu_alpha": 0}, "force_earliest_schedule": false}

2025-06-12 00:47:35.653082: I external/local_xla/xla/service/gpu/autotuning/conv_algorithm_picker.cc:549] Omitted potentially buggy algorithm eng14{k25=0} for conv %cudnn-conv-bias-activation.41 = (f32[20,128,75,75]{3,2,1,0}, u8[0]{0}) custom-call(f32[20,64,75,75]{3,2,1,0} %bitcast.3547, f32[128,64,3,3]{3,2,1,0} %bitcast.3554, f32[128]{0} %bitcast.3556), window={size=3x3 pad=1_1x1_1}, dim_labels=bf01_o1_o1->bf01, custom_call_target="__cudnn\$convBiasActivationForward", metadata={op_type="Conv2D" op_name="functional_1/vgg16_1/block2_conv1_1/convolution" source_file="/home/eliana/escola/venv/lib/python3.12/site-packages/tensorflow/python/framework/ops.py" source_line=1200}, backend_config={"operation_queue_id": "0", "wait_on_operation_queues": [], "cudnn_conv_backend_config": {"conv_result_scale": 1, "activation_mode": "kRelu", "side_input_scale": 0, "leakyrelu_alpha": 0}, "force_earliest_schedule": false}

2025-06-12 00:47:35.945630: I external/local_xla/xla/service/gpu/autotuning/conv_algorithm_picker.cc:549] Omitted potentially buggy algorithm eng14{k25=0} for conv %cudnn-conv-bias-activation.42 = (f32[20,128,75,75]{3,2,1,0}, u8[0]{0}) custom-call(f32[20,128,75,75]{3,2,1,0} %bitcast.3561, f32[128,128,3,3]{3,2,1,0} %bitcast.3568, f32[128]{0} %bitcast.3570), window={size=3x3 pad=1_1x1_1}, dim_labels=bf01_o1_o1->bf01, custom_call_target="__cudnn\$convBiasActivationForward", metadata={op_type="Conv2D" op_name="functional_1/vgg16_1/block2_conv2_1/convolution" source_file="/home/eliana/escola/venv/lib/python3.12/site-packages/tensorflow/python/framework/ops.py" source_line=1200}, backend_config={"operation_queue_id": "0", "wait_on_operation_queues": [], "cudnn_conv_backend_config": {"conv_result_scale": 1, "activation_mode": "kRelu", "side_input_scale": 0, "leakyrelu_alpha": 0}, "force_earliest_schedule": false}

2025-06-12 00:47:36.447097: I external/local_xla/xla/service/gpu/autotuning/conv_algorithm_picker.cc:549] Omitted potentially buggy algorithm eng14{k25=0} for conv %cudnn-conv-bias-activation.43 = (f32[20,256,37,37]{3,2,1,0}, u8[0]{0}) custom-call(f32[20,128,37,37]{3,2,1,0} %bitcast.3576, f32[256,128,3,3]{3,2,1,0} %bitcast.3583, f32[256]{0} %bitcast.3585), window={size=3x3 pad=1_1x1_1}, dim_labels=bf01_o1_o1->bf01, custom_call_target="__cudnn\$convBiasActivationForward", metadata={op_type="Conv2D" op_name="functional_1/vgg16_1/block3_conv1_1/convolution" source_file="/home/eliana/escola/venv/lib/python3.12/site-packages/tensorflow/python/framework/ops.py" source_line=1200}, backend_config={"operation_queue_id": "0", "wait_on_operation_queues": [], "cudnn_conv_backend_config": {"conv_result_scale": 1, "activation_mode": "kRelu", "side_input_scale": 0, "leakyrelu_alpha": 0}, "force_earliest_schedule": false}

```

2025-06-12 00:47:36.671924: I external/local_xla/xla/service/gpu/autotuning/conv_algorithm_picker.cc:549] Omitted potentially buggy algorithm eng14{k25=0} for conv %cudnn-conv-bias-activation.44 = (f32[20,256,37,37]{3,2,1,0}, u8[0]{0}) custom-call(f32[20,256,37,37]{3,2,1,0} %bitcast.3590, f32[256,256,3,3]{3,2,1,0} %bitcast.3597, f32[256]{0} %bitcast.3599), window={size=3x3 pad=1_1x1_1}, dim_labels=bf01_o101->bf01, custom_call_target="__cudnn$convBiasActivationForward", metadata={op_type="Conv2D" op_name="functional_1/vgg16_1/block3_conv2_1/convolution" source_file="/home/eliana/escola/venv/lib/python3.12/site-packages/tensorflow/python/framework/ops.py" source_line=1200}, backend_config={"operation_queue_id": "0", "wait_on_operation_queues": [], "cudnn_conv_backend_config": {"conv_result_scale": 1, "activation_mode": "kRelu", "side_input_scale": 0, "leakyrelu_alpha": 0}, "force_earliest_schedule": false}

2025-06-12 00:47:37.046567: I external/local_xla/xla/service/gpu/autotuning/conv_algorithm_picker.cc:549] Omitted potentially buggy algorithm eng14{k25=0} for conv %cudnn-conv-bias-activation.46 = (f32[20,512,18,18]{3,2,1,0}, u8[0]{0}) custom-call(f32[20,256,18,18]{3,2,1,0} %bitcast.3619, f32[512,256,3,3]{3,2,1,0} %bitcast.3626, f32[512]{0} %bitcast.3628), window={size=3x3 pad=1_1x1_1}, dim_labels=bf01_o101->bf01, custom_call_target="__cudnn$convBiasActivationForward", metadata={op_type="Conv2D" op_name="functional_1/vgg16_1/block4_conv1_1/convolution" source_file="/home/eliana/escola/venv/lib/python3.12/site-packages/tensorflow/python/framework/ops.py" source_line=1200}, backend_config={"operation_queue_id": "0", "wait_on_operation_queues": [], "cudnn_conv_backend_config": {"conv_result_scale": 1, "activation_mode": "kRelu", "side_input_scale": 0, "leakyrelu_alpha": 0}, "force_earliest_schedule": false}

2025-06-12 00:47:37.269048: I external/local_xla/xla/service/gpu/autotuning/conv_algorithm_picker.cc:549] Omitted potentially buggy algorithm eng14{k25=0} for conv %cudnn-conv-bias-activation.47 = (f32[20,512,18,18]{3,2,1,0}, u8[0]{0}) custom-call(f32[20,512,18,18]{3,2,1,0} %bitcast.3633, f32[512,512,3,3]{3,2,1,0} %bitcast.3640, f32[512]{0} %bitcast.3642), window={size=3x3 pad=1_1x1_1}, dim_labels=bf01_o101->bf01, custom_call_target="__cudnn$convBiasActivationForward", metadata={op_type="Conv2D" op_name="functional_1/vgg16_1/block4_conv2_1/convolution" source_file="/home/eliana/escola/venv/lib/python3.12/site-packages/tensorflow/python/framework/ops.py" source_line=1200}, backend_config={"operation_queue_id": "0", "wait_on_operation_queues": [], "cudnn_conv_backend_config": {"conv_result_scale": 1, "activation_mode": "kRelu", "side_input_scale": 0, "leakyrelu_alpha": 0}, "force_earliest_schedule": false}

2025-06-12 00:47:37.676869: I external/local_xla/xla/service/gpu/autotuning/conv_algorithm_picker.cc:549] Omitted potentially buggy algorithm eng14{k25=0} for conv %cudnn-conv-bias-activation.49 = (f32[20,512,9,9]{3,2,1,0}, u8[0]{0}) custom-call(f32[20,512,9,9]{3,2,1,0} %bitcast.3662, f32[512,512,3,3]{3,2,1,0} %bitcast.3669, f32[512]{0} %bitcast.3671), window={size=3x3 pad=1_1x1_1}, dim_labels=bf01_o101->bf01, custom_call_target="__cudnn$convBiasActivationForward", metadata={op_type="Conv2D" op_name="functional_1/vgg16_1/block5_conv1_1/convolution" source_file="/home/eliana/escola/venv/lib/python3.12/site-packages/tensorflow/python/framework/ops.py" source_line=1200}, backend_config={"operation_queue_id": "0", "wait_on_operation_queues": [], "cudnn_conv_backend_config": {"conv_result_scale": 1, "activation_mode": "kRelu", "side_input_scale": 0, "leakyrelu_alpha": 0}, "force_earliest_schedule": false}

```

134/134 ━━━━━━ 0s 137ms/step - accuracy: 0.3731 - loss: 12.9672

```

2025-06-12 00:47:44.211998: I external/local_xla/xla/service/gpu/autotuning/conv_algorithm_picker.cc:549] Omitted potentially buggy algorithm eng14{k25=0} for conv %cudnn-conv-bias-activation.39 = (f32[12,64,150,150]{3,2,1,0}, u8[0]{0}) custom-call(f32[12,3,150,150]{3,2,1,0} %bitcast.850, f32[64,3,3,3]{3,2,1,0} %bitcast.857, f32[64]{0} %bitcast.859), window={size=3x3 pad=1_1x1_1}, dim_labels=bf01_oi01->bf01, custom_call_target="__cudnn$convBiasActivationForward", metadata={op_type="Conv2D" op_name="functional_1/vgg16_1/block1_conv1_1/convolution" source_file="/home/eliana/escola/venv/lib/python3.12/site-packages/tensorflow/python/framework/ops.py" source_line=1200}, backend_config={"operation_queue_id":"0","wait_on_operation_queues":[],"cudnn_conv_backend_config":{"conv_result_scale":1,"activation_mode":"kRelu","side_input_scale":0,"leakyrelu_alpha":0},"force_earliest_schedule":false}

2025-06-12 00:47:44.268708: I external/local_xla/xla/service/gpu/autotuning/conv_algorithm_picker.cc:549] Omitted potentially buggy algorithm eng14{k25=0} for conv %cudnn-conv-bias-activation.40 = (f32[12,64,150,150]{3,2,1,0}, u8[0]{0}) custom-call(f32[12,64,150,150]{3,2,1,0} %bitcast.864, f32[64,64,3,3]{3,2,1,0} %bitcast.871, f32[64]{0} %bitcast.873), window={size=3x3 pad=1_1x1_1}, dim_labels=bf01_oi01->bf01, custom_call_target="__cudnn$convBiasActivationForward", metadata={op_type="Conv2D" op_name="functional_1/vgg16_1/block1_conv2_1/convolution" source_file="/home/eliana/escola/venv/lib/python3.12/site-packages/tensorflow/python/framework/ops.py" source_line=1200}, backend_config={"operation_queue_id":"0","wait_on_operation_queues":[],"cudnn_conv_backend_config":{"conv_result_scale":1,"activation_mode":"kRelu","side_input_scale":0,"leakyrelu_alpha":0},"force_earliest_schedule":false}

2025-06-12 00:47:44.696763: I external/local_xla/xla/service/gpu/autotuning/conv_algorithm_picker.cc:549] Omitted potentially buggy algorithm eng14{k25=0} for conv %cudnn-conv-bias-activation.41 = (f32[12,128,75,75]{3,2,1,0}, u8[0]{0}) custom-call(f32[12,64,75,75]{3,2,1,0} %bitcast.881, f32[128,64,3,3]{3,2,1,0} %bitcast.888, f32[128]{0} %bitcast.890), window={size=3x3 pad=1_1x1_1}, dim_labels=bf01_oi01->bf01, custom_call_target="__cudnn$convBiasActivationForward", metadata={op_type="Conv2D" op_name="functional_1/vgg16_1/block2_conv1_1/convolution" source_file="/home/eliana/escola/venv/lib/python3.12/site-packages/tensorflow/python/framework/ops.py" source_line=1200}, backend_config={"operation_queue_id":"0","wait_on_operation_queues":[],"cudnn_conv_backend_config":{"conv_result_scale":1,"activation_mode":"kRelu","side_input_scale":0,"leakyrelu_alpha":0},"force_earliest_schedule":false}

2025-06-12 00:47:44.901823: I external/local_xla/xla/service/gpu/autotuning/conv_algorithm_picker.cc:549] Omitted potentially buggy algorithm eng14{k25=0} for conv %cudnn-conv-bias-activation.42 = (f32[12,128,75,75]{3,2,1,0}, u8[0]{0}) custom-call(f32[12,128,75,75]{3,2,1,0} %bitcast.895, f32[128,128,3,3]{3,2,1,0} %bitcast.902, f32[128]{0} %bitcast.904), window={size=3x3 pad=1_1x1_1}, dim_labels=bf01_oi01->bf01, custom_call_target="__cudnn$convBiasActivationForward", metadata={op_type="Conv2D" op_name="functional_1/vgg16_1/block2_conv2_1/convolution" source_file="/home/eliana/escola/venv/lib/python3.12/site-packages/tensorflow/python/framework/ops.py" source_line=1200}, backend_config={"operation_queue_id":"0","wait_on_operation_queues":[],"cudnn_conv_backend_config":{"conv_result_scale":1,"activation_mode":"kRelu","side_input_scale":0,"leakyrelu_alpha":0},"force_earliest_schedule":false}

2025-06-12 00:47:45.260170: I external/local_xla/xla/service/gpu/autotuning/conv_algorithm_picker.cc:549] Omitted potentially buggy algorithm eng14{k25=0} for conv %cudnn-conv-bias-activation.43 = (f32[12,256,37,37]{3,2,1,0}, u8[0]{0}) custom-call(f32[12,128,37,37]{3,2,1,0} %bitcast.910, f32[256,128,3,3]{3,2,1,0} %bitcast.917, f32[256]{0} %bitcast.919), window={size=3x3 pad=1_1x1_1}, dim_labels=bf01_oi01->bf01, custom_call_target="__cudnn$convBiasActivationForward", metadata={op_type="Conv2D" op_name="functional_1/vgg16_1/block3_conv1_1/convolution" source_file="/home/eliana/escola/venv/lib/python3.12/site-packages/tensorflow/python/framework/ops.py" source_line=1200}, backend_config={"operation_queue_id":"0","wait_on_operation_queues":[],"cudnn_conv_backend_config":{"conv_result_scale":1,"activation_mode":"kRelu","side_input_scale":0,"leakyrelu_alpha":0},"force_earliest_schedule":false}

```

2025-06-12 00:47:45.431461: I external/local_xla/xla/service/gpu/autotuning/conv_algorithm_picker.cc:549] Omitted potentially buggy algorithm eng14{k25=0} for conv %cudnn-conv-bias-activation.44 = (f32[12,256,37,37]{3,2,1,0}, u8[0]{0}) custom-call(f32[12,256,37,37]{3,2,1,0} %bitcast.924, f32[256,256,3,3]{3,2,1,0} %bitcast.931, f32[256]{0} %bitcast.933), window={size=3x3 pad=1_1x1_1}, dim_labels=bf01_oi01->bf01, custom_call_target="__cudnn\$convBiasActivationForward", metadata={op_type="Conv2D" op_name="functional_1/vgg16_1/block3_conv2_1/convolution" source_file="/home/eliana/escola/venv/lib/python3.12/site-packages/tensorflow/python/framework/ops.py" source_line=1200}, backend_config={"operation_queue_id": "0", "wait_on_operation_queues": [], "cudnn_conv_backend_config": {"conv_result_scale": 1, "activation_mode": "kRelu", "side_input_scale": 0, "leakyrelu_alpha": 0}, "force_earliest_schedule": false}

2025-06-12 00:47:45.703435: I external/local_xla/xla/service/gpu/autotuning/conv_algorithm_picker.cc:549] Omitted potentially buggy algorithm eng14{k25=0} for conv %cudnn-conv-bias-activation.46 = (f32[12,512,18,18]{3,2,1,0}, u8[0]{0}) custom-call(f32[12,256,18,18]{3,2,1,0} %bitcast.953, f32[512,256,3,3]{3,2,1,0} %bitcast.960, f32[512]{0} %bitcast.962), window={size=3x3 pad=1_1x1_1}, dim_labels=bf01_oi01->bf01, custom_call_target="__cudnn\$convBiasActivationForward", metadata={op_type="Conv2D" op_name="functional_1/vgg16_1/block4_conv1_1/convolution" source_file="/home/eliana/escola/venv/lib/python3.12/site-packages/tensorflow/python/framework/ops.py" source_line=1200}, backend_config={"operation_queue_id": "0", "wait_on_operation_queues": [], "cudnn_conv_backend_config": {"conv_result_scale": 1, "activation_mode": "kRelu", "side_input_scale": 0, "leakyrelu_alpha": 0}, "force_earliest_schedule": false}

2025-06-12 00:47:45.865796: I external/local_xla/xla/service/gpu/autotuning/conv_algorithm_picker.cc:549] Omitted potentially buggy algorithm eng14{k25=0} for conv %cudnn-conv-bias-activation.47 = (f32[12,512,18,18]{3,2,1,0}, u8[0]{0}) custom-call(f32[12,512,18,18]{3,2,1,0} %bitcast.967, f32[512,512,3,3]{3,2,1,0} %bitcast.974, f32[512]{0} %bitcast.976), window={size=3x3 pad=1_1x1_1}, dim_labels=bf01_oi01->bf01, custom_call_target="__cudnn\$convBiasActivationForward", metadata={op_type="Conv2D" op_name="functional_1/vgg16_1/block4_conv2_1/convolution" source_file="/home/eliana/escola/venv/lib/python3.12/site-packages/tensorflow/python/framework/ops.py" source_line=1200}, backend_config={"operation_queue_id": "0", "wait_on_operation_queues": [], "cudnn_conv_backend_config": {"conv_result_scale": 1, "activation_mode": "kRelu", "side_input_scale": 0, "leakyrelu_alpha": 0}, "force_earliest_schedule": false}

2025-06-12 00:47:46.155778: I external/local_xla/xla/service/gpu/autotuning/conv_algorithm_picker.cc:549] Omitted potentially buggy algorithm eng14{k25=0} for conv %cudnn-conv-bias-activation.49 = (f32[12,512,9,9]{3,2,1,0}, u8[0]{0}) custom-call(f32[12,512,9,9]{3,2,1,0} %bitcast.996, f32[512,512,3,3]{3,2,1,0} %bitcast.1003, f32[512]{0} %bitcast.1005), window={size=3x3 pad=1_1x1_1}, dim_labels=bf01_oi01->bf01, custom_call_target="__cudnn\$convBiasActivationForward", metadata={op_type="Conv2D" op_name="functional_1/vgg16_1/block5_conv1_1/convolution" source_file="/home/eliana/escola/venv/lib/python3.12/site-packages/tensorflow/python/framework/ops.py" source_line=1200}, backend_config={"operation_queue_id": "0", "wait_on_operation_queues": [], "cudnn_conv_backend_config": {"conv_result_scale": 1, "activation_mode": "kRelu", "side_input_scale": 0, "leakyrelu_alpha": 0}, "force_earliest_schedule": false}

```
134/134 ━━━━━━━━━━ 33s 197ms/step - accuracy: 0.3736 - loss: 12.9125 -  
val_accuracy: 0.5261 - val_loss: 1.5430  
Epoch 2/8  
134/134 ━━━━━━━━━━ 19s 142ms/step - accuracy: 0.5521 - loss: 1.5502 - v  
al_accuracy: 0.5993 - val_loss: 1.5276  
Epoch 3/8  
134/134 ━━━━━━━━━━ 19s 142ms/step - accuracy: 0.6292 - loss: 1.3398 - v  
al_accuracy: 0.6563 - val_loss: 1.2842  
Epoch 4/8  
134/134 ━━━━━━━━━━ 19s 143ms/step - accuracy: 0.6726 - loss: 1.1242 - v  
al_accuracy: 0.6415 - val_loss: 1.4823  
Epoch 5/8  
134/134 ━━━━━━━━━━ 19s 143ms/step - accuracy: 0.6984 - loss: 1.0145 - v  
al_accuracy: 0.6613 - val_loss: 1.3911  
Epoch 6/8  
134/134 ━━━━━━━━━━ 19s 143ms/step - accuracy: 0.7314 - loss: 0.8690 - v  
al_accuracy: 0.6514 - val_loss: 1.4735  
Epoch 7/8  
134/134 ━━━━━━━━━━ 19s 144ms/step - accuracy: 0.7759 - loss: 0.7751 - v  
al_accuracy: 0.6627 - val_loss: 1.6501  
Epoch 8/8  
134/134 ━━━━━━━━━━ 19s 144ms/step - accuracy: 0.7959 - loss: 0.7172 - v  
al_accuracy: 0.6951 - val_loss: 1.6270
```

```
In [10]: print_classification_metrics(model_t, test_dataset, "Modelo 2 : VGG16 (Feature E")
```

1/1 ————— 1s 613ms/step
 1/1 ————— 0s 129ms/step
 1/1 ————— 0s 128ms/step
 1/1 ————— 0s 128ms/step
 1/1 ————— 0s 134ms/step
 1/1 ————— 0s 127ms/step
 1/1 ————— 0s 127ms/step
 1/1 ————— 0s 126ms/step
 1/1 ————— 0s 128ms/step
 1/1 ————— 0s 129ms/step
 1/1 ————— 0s 129ms/step
 1/1 ————— 0s 128ms/step
 1/1 ————— 0s 136ms/step
 1/1 ————— 0s 126ms/step
 1/1 ————— 0s 128ms/step
 1/1 ————— 0s 126ms/step
 1/1 ————— 0s 130ms/step
 1/1 ————— 0s 127ms/step
 1/1 ————— 0s 127ms/step
 1/1 ————— 0s 129ms/step
 1/1 ————— 0s 135ms/step
 1/1 ————— 0s 126ms/step
 1/1 ————— 0s 125ms/step
 1/1 ————— 0s 127ms/step
 1/1 ————— 0s 127ms/step
 1/1 ————— 0s 127ms/step
 1/1 ————— 0s 127ms/step
 1/1 ————— 0s 128ms/step
 1/1 ————— 0s 129ms/step
 1/1 ————— 0s 127ms/step
 1/1 ————— 0s 128ms/step
 1/1 ————— 0s 128ms/step
 1/1 ————— 0s 125ms/step
 1/1 ————— 0s 131ms/step
 1/1 ————— 0s 127ms/step
 1/1 ————— 0s 125ms/step
 1/1 ————— 0s 128ms/step
 1/1 ————— 0s 127ms/step
 1/1 ————— 0s 126ms/step
 1/1 ————— 0s 125ms/step
 1/1 ————— 0s 129ms/step
 1/1 ————— 0s 127ms/step
 1/1 ————— 0s 128ms/step
 1/1 ————— 0s 127ms/step
 1/1 ————— 1s 538ms/step

Modelo 2 : VGG16 (Feature Extraction com Augmentation)

	precision	recall	f1-score	support
--	-----------	--------	----------	---------

0	0.6364	0.7179	0.6747	195
1	0.7081	0.6616	0.6841	198
2	0.5692	0.6575	0.6102	219
3	0.9191	0.8281	0.8712	192
4	0.6010	0.5631	0.5814	222
5	0.7347	0.7200	0.7273	200
6	0.9730	0.9278	0.9499	194

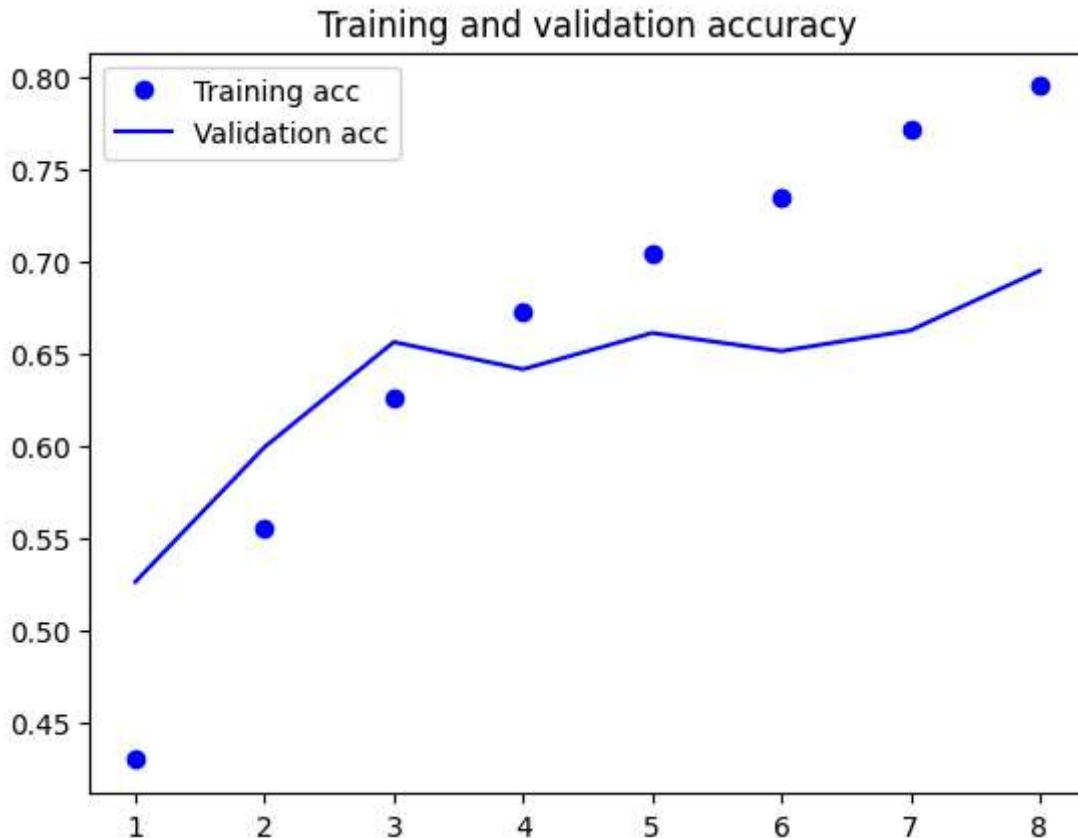
accuracy			0.7204	1420
macro avg	0.7345	0.7252	0.7284	1420

weighted avg	0.7285	0.7204	0.7230	1420
--------------	--------	--------	--------	------

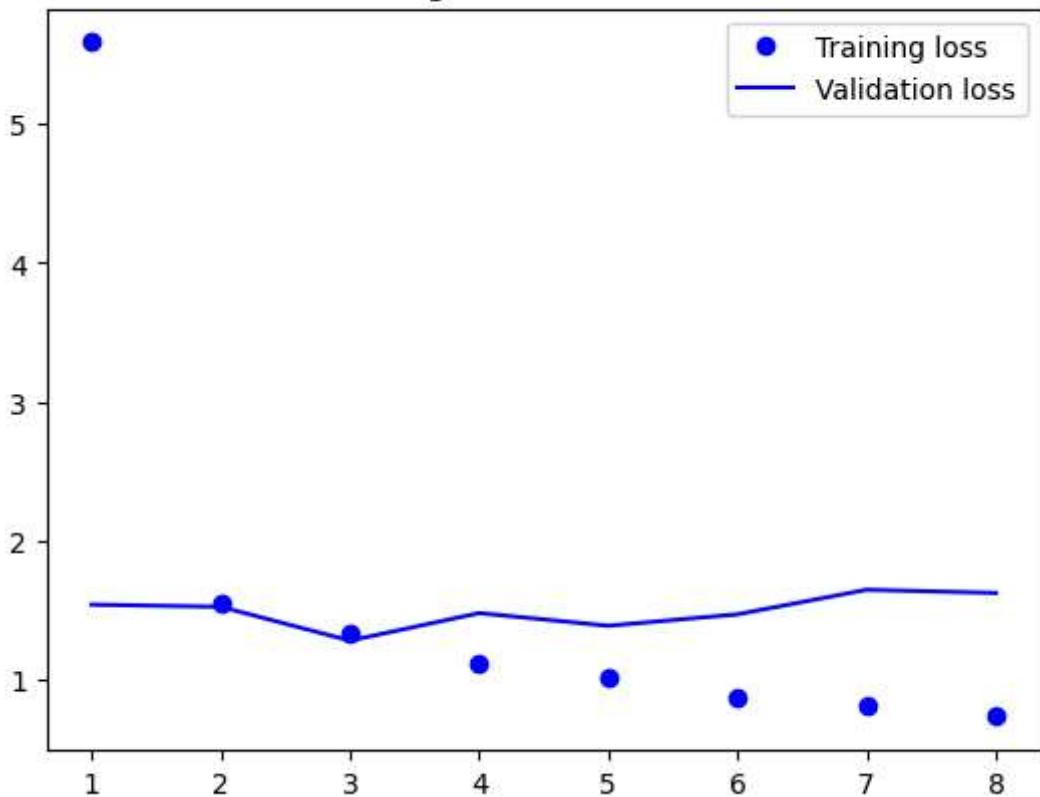
2025-06-12 00:50:08.781264: I tensorflow/core/framework/local_rendezvous.cc:407] Local rendezvous is aborting with status: OUT_OF_RANGE: End of sequence

Curvas de Loss e Accuracy

```
In [11]: import matplotlib.pyplot as plt
accuracy = history_t.history['accuracy']
val_acc = history_t.history['val_accuracy']
loss = history_t.history['loss']
val_loss = history_t.history['val_loss']
epochs = range(1, len(accuracy) + 1)
plt.plot(epochs, accuracy, 'bo', label='Training acc')
plt.plot(epochs, val_acc, 'b', label='Validation acc')
plt.title('Training and validation accuracy')
plt.legend()
plt.figure()
plt.plot(epochs, loss, 'bo', label='Training loss')
plt.plot(epochs, val_loss, 'b', label='Validation loss')
plt.title('Training and validation loss')
plt.legend()
plt.show()
```



Training and validation loss



A acurácia de treino aumenta rapidamente até cerca de 0.80, enquanto a acurácia de validação também melhora, alcançando aproximadamente 0.70. As curvas mantêm-se relativamente próximas, o que sugere bom aprendizado inicial sem sinais evidentes de overfitting, embora o número de épocas ainda seja pequeno para uma conclusão definitiva.

A loss de treino diminui visivelmente ao longo das épocas, enquanto a loss de validação apresenta variações e oscilações leves, mantendo-se num patamar entre 1.3 e 1.7. Apesar da tendência geral ser decrescente, estas flutuações indicam que o modelo ainda está a ajustar-se à tarefa, e que seria útil continuar o treino por mais épocas para avaliar se a tendência se mantém.

Matriz de Confusão

```
In [12]: import numpy as np
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
import matplotlib.pyplot as plt

# Obter previsões no test_dataset
y_true = []
y_pred = []

for images, labels in test_dataset:
    preds = model_t.predict(images)
    y_true.extend(np.argmax(labels.numpy(), axis=1))
    y_pred.extend(np.argmax(preds, axis=1))

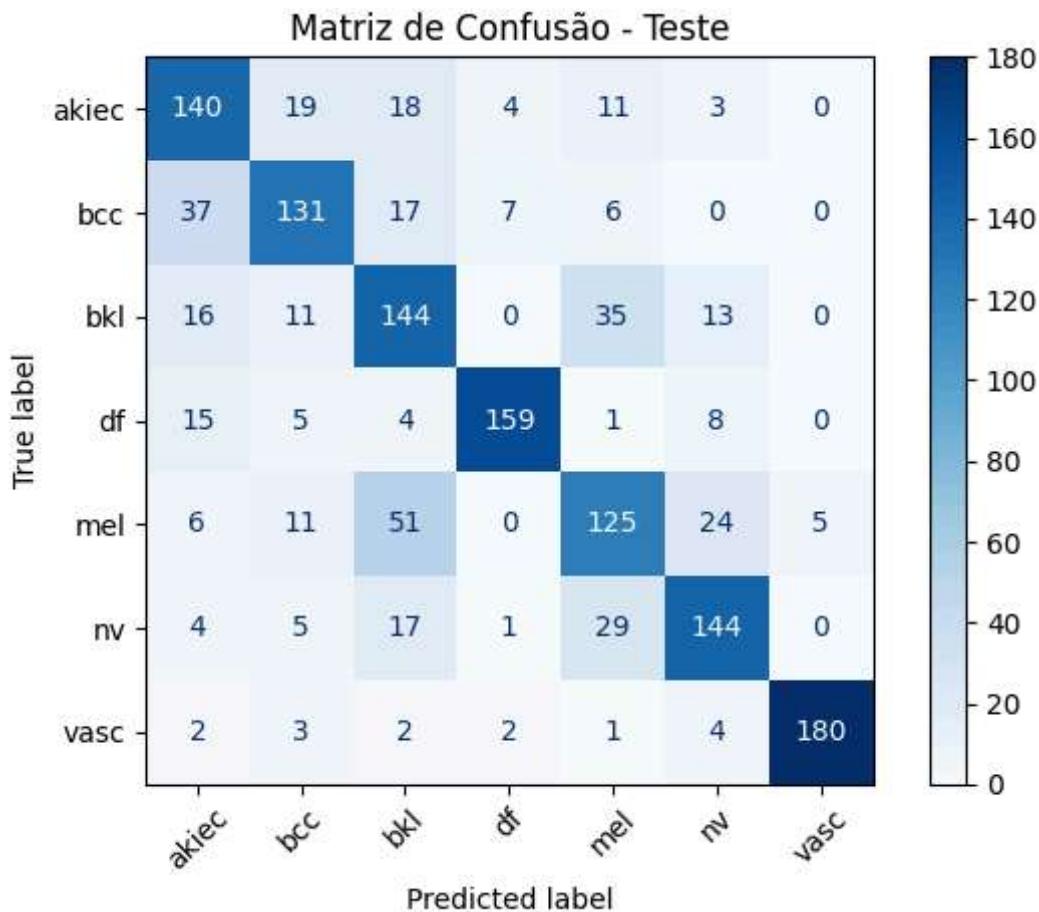
class_names = test_dataset.class_names
```

```
# Criar e mostrar a matriz de confusão
cm = confusion_matrix(y_true, y_pred)
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=class_names)

plt.figure(figsize=(10, 8))
disp.plot(cmap=plt.cm.Blues, xticks_rotation=45, values_format='d')
plt.title("Matriz de Confusão - Teste")
plt.tight_layout()
plt.show()
```

1/1 ━━━━━━ 0s 134ms/step
 1/1 ━━━━━━ 0s 130ms/step
 1/1 ━━━━━━ 0s 133ms/step
 1/1 ━━━━━━ 0s 128ms/step
 1/1 ━━━━━━ 0s 129ms/step
 1/1 ━━━━━━ 0s 128ms/step
 1/1 ━━━━━━ 0s 129ms/step
 1/1 ━━━━━━ 0s 129ms/step
 1/1 ━━━━━━ 0s 128ms/step
 1/1 ━━━━━━ 0s 127ms/step
 1/1 ━━━━━━ 0s 127ms/step
 1/1 ━━━━━━ 0s 129ms/step
 1/1 ━━━━━━ 0s 126ms/step
 1/1 ━━━━━━ 0s 127ms/step
 1/1 ━━━━━━ 0s 128ms/step
 1/1 ━━━━━━ 0s 127ms/step
 1/1 ━━━━━━ 0s 126ms/step
 1/1 ━━━━━━ 0s 127ms/step
 1/1 ━━━━━━ 0s 134ms/step
 1/1 ━━━━━━ 0s 126ms/step
 1/1 ━━━━━━ 0s 129ms/step
 1/1 ━━━━━━ 0s 127ms/step
 1/1 ━━━━━━ 0s 126ms/step
 1/1 ━━━━━━ 0s 128ms/step
 1/1 ━━━━━━ 0s 127ms/step
 1/1 ━━━━━━ 0s 128ms/step
 1/1 ━━━━━━ 0s 130ms/step
 1/1 ━━━━━━ 0s 129ms/step
 1/1 ━━━━━━ 0s 127ms/step
 1/1 ━━━━━━ 0s 128ms/step
 1/1 ━━━━━━ 0s 127ms/step
 1/1 ━━━━━━ 0s 126ms/step
 1/1 ━━━━━━ 0s 132ms/step
 1/1 ━━━━━━ 0s 127ms/step
 1/1 ━━━━━━ 0s 129ms/step
 1/1 ━━━━━━ 0s 127ms/step
 1/1 ━━━━━━ 0s 130ms/step
 1/1 ━━━━━━ 0s 127ms/step
 1/1 ━━━━━━ 0s 127ms/step
 1/1 ━━━━━━ 0s 127ms/step
 1/1 ━━━━━━ 0s 128ms/step
 1/1 ━━━━━━ 0s 127ms/step
 1/1 ━━━━━━ 0s 128ms/step
 1/1 ━━━━━━ 0s 64ms/step

2025-06-12 00:50:15.968487: I tensorflow/core/framework/local_rendezvous.cc:407] Local rendezvous is aborting with status: OUT_OF_RANGE: End of sequence
 <Figure size 1000x800 with 0 Axes>



O modelo apresenta bom desempenho geral, com elevada taxa de acertos nas classes 'akiec', 'df', 'nv' e 'vasc'. No entanto, observa-se confusão significativa entre as classes 'mel', 'bkl' e 'nv', o que pode indicar dificuldade do modelo em distinguir lesões visualmente semelhantes. Ainda assim, o padrão de acertos e erros é consistente com os desafios esperados em tarefas de classificação de imagens dermatológicas.

Fine Tuning

```
In [13]: # Descongelar parte da VGG16 (últimas camadas)
conv_base.trainable = True
for layer in conv_base.layers[:-4]: # Descongelar as ultimas 4 camadas
    layer.trainable = False # manter as primeiras camadas congeladas

# Compilar o modelo com Learning rate
model_t.compile(
    loss='categorical_crossentropy',
    optimizer=keras.optimizers.SGD(learning_rate=3.449033551761605e-05),
    metrics=['accuracy']
)

# Treinar novamente (ajustar a parte da VGG16)
history_t = model_t.fit(
    train_dataset,
    validation_data=validation_dataset,
    epochs=20
)
```

```
#history = model_t.fit(
    #train_dataset, #Inicia o treino do modelo usando o conjunto de dados de treino
    #epochs=30, # Apesar de ter um numero alto aqui, o EarlyStopping para automaticamente
    #validation_data=validation_dataset # Usa o conjunto de dados de validação para parar o treino
    #, callbacks=callbacks_list
#)
```

Epoch 1/20

```
2025-06-12 00:50:17.258859: I external/local_xla/xla/service/gpu/autotuning/conv_algorithm_picker.cc:549] Omitted potentially buggy algorithm eng14{k25=0} for convolution %cudnn-conv-bias-activation.49 = (f32[32,512,9,9]{3,2,1,0}, u8[0]{0}) custom-call(f32[32,512,9,9]{3,2,1,0} %bitcast.4353, f32[512,512,3,3]{3,2,1,0} %bitcast.4173, f32[512]{0} %bitcast.4517), window={size=3x3 pad=1_1x1_1}, dim_labels=bf01_o01->bf01, custom_call_target="__cudnn$convBiasActivationForward", metadata={op_type="Conv2D" op_name="functional_1/vgg16_1/block5_conv1_1/convolution" source_file="/home/eliana/escola/venv/lib/python3.12/site-packages/tensorflow/python/framework/ops.py" source_line=1200}, backend_config={"operation_queue_id": "0", "wait_on_operation_queues": [], "cudnn_conv_backend_config": {"conv_result_scale": 1, "activation_mode": "kNone", "side_input_scale": 0, "leakyrelu_alpha": 0}, "force_earliest_schedule": false}
```

133/134 ————— 0s 122ms/step - accuracy: 0.8234 - loss: 0.6356

```
2025-06-12 00:50:35.423069: I external/local_xla/xla/service/gpu/autotuning/conv_algorithm_picker.cc:549] Omitted potentially buggy algorithm eng14{k25=0} for convolution %cudnn-conv-bias-activation.49 = (f32[20,512,9,9]{3,2,1,0}, u8[0]{0}) custom-call(f32[20,512,9,9]{3,2,1,0} %bitcast.4353, f32[512,512,3,3]{3,2,1,0} %bitcast.4173, f32[512]{0} %bitcast.4517), window={size=3x3 pad=1_1x1_1}, dim_labels=bf01_o01->bf01, custom_call_target="__cudnn$convBiasActivationForward", metadata={op_type="Conv2D" op_name="functional_1/vgg16_1/block5_conv1_1/convolution" source_file="/home/eliana/escola/venv/lib/python3.12/site-packages/tensorflow/python/framework/ops.py" source_line=1200}, backend_config={"operation_queue_id": "0", "wait_on_operation_queues": [], "cudnn_conv_backend_config": {"conv_result_scale": 1, "activation_mode": "kNone", "side_input_scale": 0, "leakyrelu_alpha": 0}, "force_earliest_schedule": false}
```

```
134/134 ━━━━━━━━━━ 27s 179ms/step - accuracy: 0.8235 - loss: 0.6347 - v  
al_accuracy: 0.7042 - val_loss: 1.5173  
Epoch 2/20  
134/134 ━━━━━━━━━━ 21s 159ms/step - accuracy: 0.8376 - loss: 0.5003 - v  
al_accuracy: 0.6993 - val_loss: 1.4958  
Epoch 3/20  
134/134 ━━━━━━━━━━ 21s 159ms/step - accuracy: 0.8523 - loss: 0.4644 - v  
al_accuracy: 0.7028 - val_loss: 1.4378  
Epoch 4/20  
134/134 ━━━━━━━━━━ 21s 159ms/step - accuracy: 0.8542 - loss: 0.4492 - v  
al_accuracy: 0.6993 - val_loss: 1.4232  
Epoch 5/20  
134/134 ━━━━━━━━━━ 21s 160ms/step - accuracy: 0.8521 - loss: 0.4472 - v  
al_accuracy: 0.7056 - val_loss: 1.4145  
Epoch 6/20  
134/134 ━━━━━━━━━━ 21s 160ms/step - accuracy: 0.8640 - loss: 0.3777 - v  
al_accuracy: 0.7063 - val_loss: 1.4244  
Epoch 7/20  
134/134 ━━━━━━━━━━ 21s 160ms/step - accuracy: 0.8681 - loss: 0.3603 - v  
al_accuracy: 0.7113 - val_loss: 1.4146  
Epoch 8/20  
134/134 ━━━━━━━━━━ 22s 161ms/step - accuracy: 0.8692 - loss: 0.3601 - v  
al_accuracy: 0.7056 - val_loss: 1.3850  
Epoch 9/20  
134/134 ━━━━━━━━━━ 22s 161ms/step - accuracy: 0.8726 - loss: 0.3757 - v  
al_accuracy: 0.7042 - val_loss: 1.3865  
Epoch 10/20  
134/134 ━━━━━━━━━━ 22s 161ms/step - accuracy: 0.8749 - loss: 0.3512 - v  
al_accuracy: 0.7077 - val_loss: 1.3997  
Epoch 11/20  
134/134 ━━━━━━━━━━ 22s 161ms/step - accuracy: 0.8703 - loss: 0.3699 - v  
al_accuracy: 0.7099 - val_loss: 1.3786  
Epoch 12/20  
134/134 ━━━━━━━━━━ 22s 161ms/step - accuracy: 0.8859 - loss: 0.3117 - v  
al_accuracy: 0.7113 - val_loss: 1.3899  
Epoch 13/20  
134/134 ━━━━━━━━━━ 22s 161ms/step - accuracy: 0.8793 - loss: 0.3330 - v  
al_accuracy: 0.7092 - val_loss: 1.3962  
Epoch 14/20  
134/134 ━━━━━━━━━━ 22s 161ms/step - accuracy: 0.8958 - loss: 0.2804 - v  
al_accuracy: 0.7085 - val_loss: 1.3723  
Epoch 15/20  
134/134 ━━━━━━━━━━ 22s 161ms/step - accuracy: 0.8819 - loss: 0.3194 - v  
al_accuracy: 0.7134 - val_loss: 1.3716  
Epoch 16/20  
134/134 ━━━━━━━━━━ 22s 161ms/step - accuracy: 0.8923 - loss: 0.2981 - v  
al_accuracy: 0.7120 - val_loss: 1.3573  
Epoch 17/20  
134/134 ━━━━━━━━━━ 22s 161ms/step - accuracy: 0.8946 - loss: 0.2781 - v  
al_accuracy: 0.7127 - val_loss: 1.3687  
Epoch 18/20  
134/134 ━━━━━━━━━━ 22s 160ms/step - accuracy: 0.8965 - loss: 0.2977 - v  
al_accuracy: 0.7120 - val_loss: 1.3839  
Epoch 19/20  
134/134 ━━━━━━━━━━ 22s 160ms/step - accuracy: 0.9098 - loss: 0.2593 - v  
al_accuracy: 0.7155 - val_loss: 1.3806  
Epoch 20/20  
134/134 ━━━━━━━━━━ 21s 160ms/step - accuracy: 0.8909 - loss: 0.2841 - v  
al_accuracy: 0.7176 - val_loss: 1.3584
```

```
In [14]: print_classification_metrics(model_t, test_dataset, "Modelo 2 : Fine-tuning")
```

1/1 ————— 1s 614ms/step
 1/1 ————— 0s 130ms/step
 1/1 ————— 0s 127ms/step
 1/1 ————— 0s 143ms/step
 1/1 ————— 0s 132ms/step
 1/1 ————— 0s 130ms/step
 1/1 ————— 0s 130ms/step
 1/1 ————— 0s 134ms/step
 1/1 ————— 0s 137ms/step
 1/1 ————— 0s 131ms/step
 1/1 ————— 0s 128ms/step
 1/1 ————— 0s 133ms/step
 1/1 ————— 0s 128ms/step
 1/1 ————— 0s 130ms/step
 1/1 ————— 0s 129ms/step
 1/1 ————— 0s 132ms/step
 1/1 ————— 0s 130ms/step
 1/1 ————— 0s 128ms/step
 1/1 ————— 0s 131ms/step
 1/1 ————— 0s 129ms/step
 1/1 ————— 0s 130ms/step
 1/1 ————— 0s 128ms/step
 1/1 ————— 0s 131ms/step
 1/1 ————— 0s 129ms/step
 1/1 ————— 0s 130ms/step
 1/1 ————— 0s 130ms/step
 1/1 ————— 0s 129ms/step
 1/1 ————— 0s 129ms/step
 1/1 ————— 0s 129ms/step
 1/1 ————— 0s 128ms/step
 1/1 ————— 0s 130ms/step
 1/1 ————— 0s 135ms/step
 1/1 ————— 0s 129ms/step
 1/1 ————— 0s 130ms/step
 1/1 ————— 0s 130ms/step
 1/1 ————— 0s 129ms/step
 1/1 ————— 0s 129ms/step
 1/1 ————— 0s 132ms/step
 1/1 ————— 0s 131ms/step
 1/1 ————— 0s 129ms/step
 1/1 ————— 0s 131ms/step
 1/1 ————— 0s 130ms/step
 1/1 ————— 0s 128ms/step
 1/1 ————— 1s 539ms/step

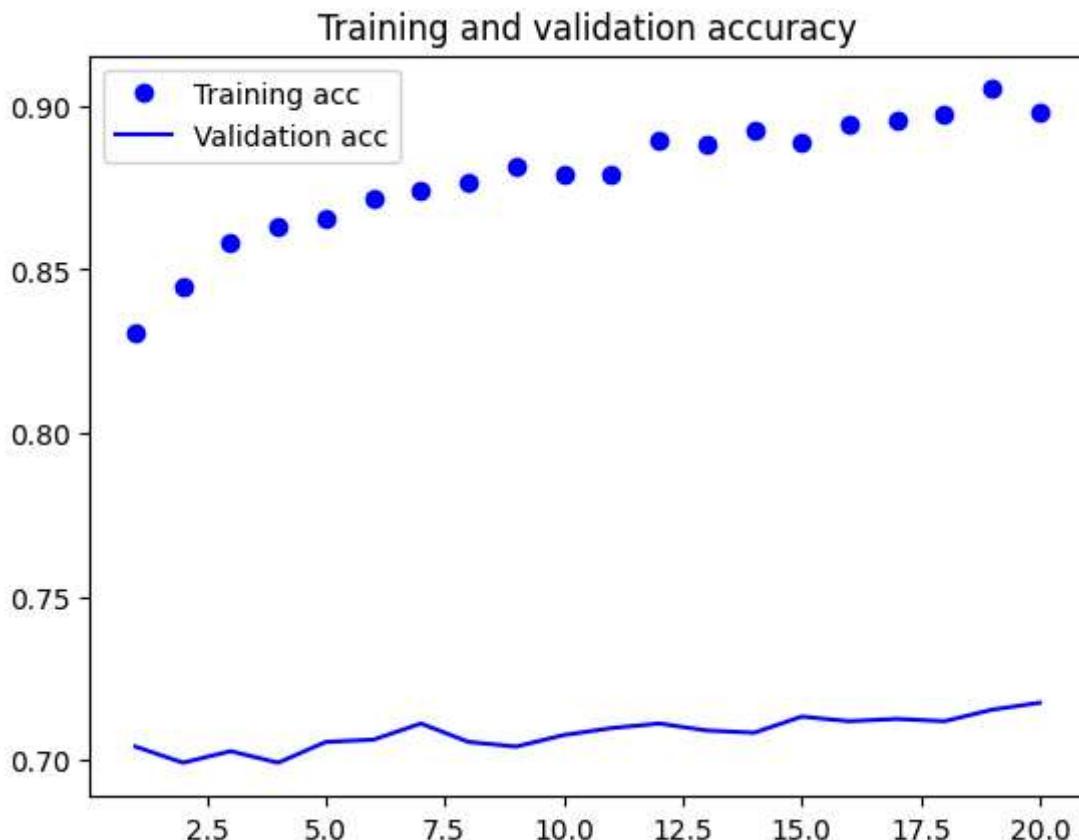
Modelo 2 : Fine-tuning

	precision	recall	f1-score	support
0	0.6991	0.7744	0.7348	195
1	0.7676	0.7172	0.7415	198
2	0.6330	0.6301	0.6316	219
3	0.8872	0.9010	0.8941	192
4	0.6311	0.5856	0.6075	222
5	0.7562	0.7600	0.7581	200
6	0.9447	0.9691	0.9567	194
accuracy			0.7563	1420
macro avg	0.7598	0.7625	0.7606	1420

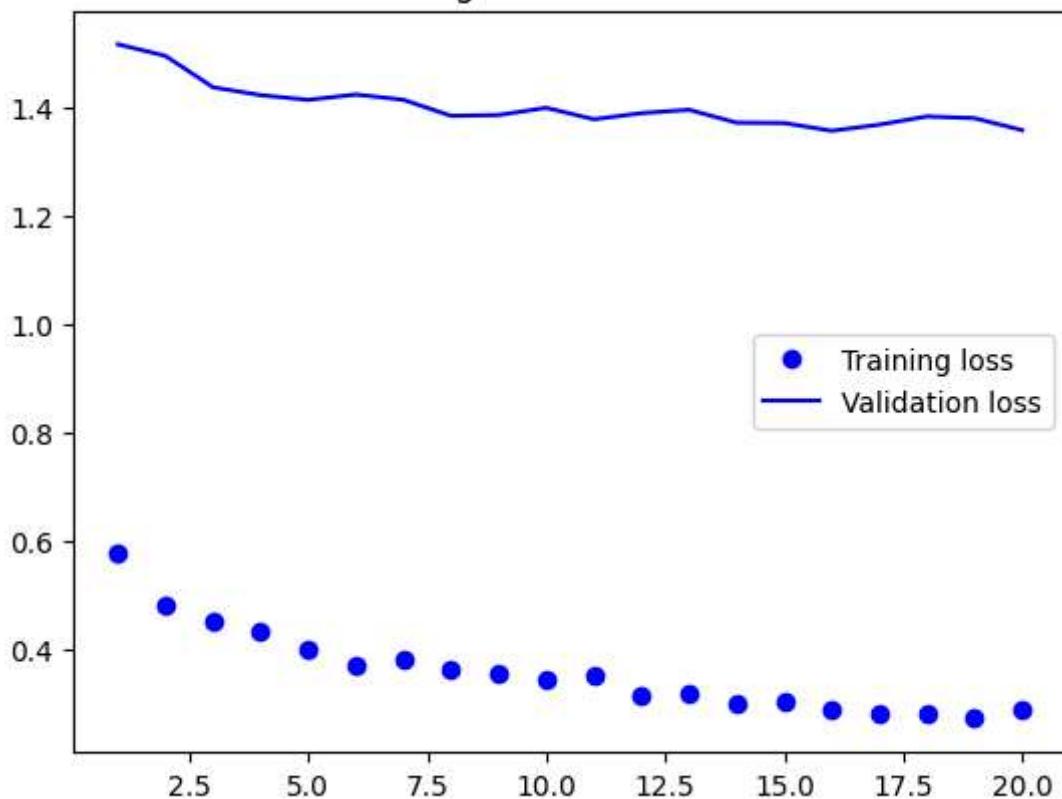
weighted avg	0.7548	0.7563	0.7550	1420
--------------	--------	--------	--------	------

Curvas de Loss e Accuracy

```
In [15]: import matplotlib.pyplot as plt
accuracy = history_t.history['accuracy']
val_acc = history_t.history['val_accuracy']
loss = history_t.history['loss']
val_loss = history_t.history['val_loss']
epochs = range(1, len(accuracy) + 1)
plt.plot(epochs, accuracy, 'bo', label='Training acc')
plt.plot(epochs, val_acc, 'b', label='Validation acc')
plt.title('Training and validation accuracy')
plt.legend()
plt.figure()
plt.plot(epochs, loss, 'bo', label='Training loss')
plt.plot(epochs, val_loss, 'b', label='Validation loss')
plt.title('Training and validation loss')
plt.legend()
plt.show()
```



Training and validation loss



A acurácia de treino aumenta de forma constante, atingindo cerca de 0.90, enquanto a acurácia de validação permanece praticamente estável em torno de 0.70. Essa discrepância crescente entre as curvas indica que o modelo está a sofrer de overfitting, ou seja, está a aprender bem os dados de treino, mas não consegue generalizar para novos dados.

A loss de treino diminui continuamente, enquanto a loss de validação permanece alta e praticamente inalterada ao longo das épocas. Isso reforça o diagnóstico de overfitting e sugere que o modelo está a memorizar os exemplos de treino sem conseguir melhorar no conjunto de validação.

Matriz de Confusão

```
In [16]: import numpy as np
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
import matplotlib.pyplot as plt

# Obter previsões no test_dataset
y_true = []
y_pred = []

for images, labels in test_dataset:
    preds = model_t.predict(images)
    y_true.extend(np.argmax(labels.numpy(), axis=1))
    y_pred.extend(np.argmax(preds, axis=1))

class_names = test_dataset.class_names

# Criar e mostrar a matriz de confusão
cm = confusion_matrix(y_true, y_pred)
```

```

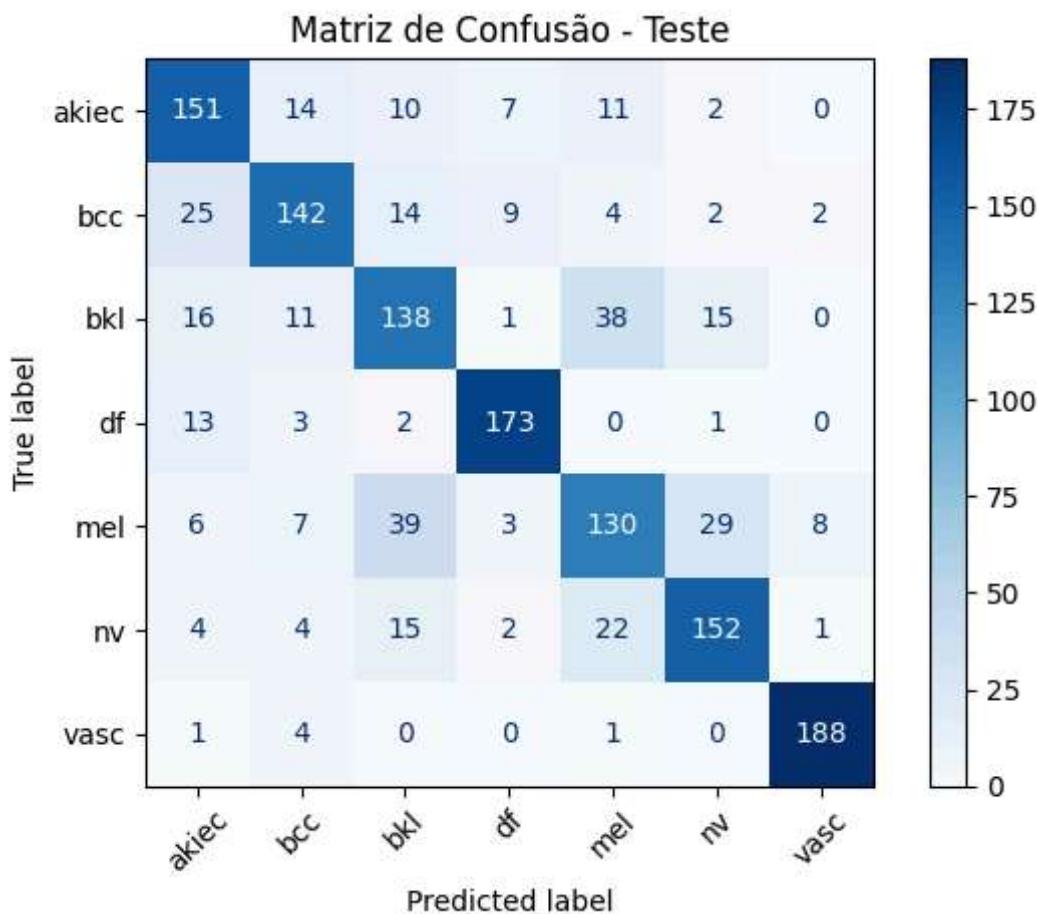
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=class_names)

plt.figure(figsize=(10, 8))
disp.plot(cmap=plt.cm.Blues, xticks_rotation=45, values_format='d')
plt.title("Matriz de Confusão - Teste")
plt.tight_layout()
plt.show()

```

1/1 0s 139ms/step
 1/1 0s 129ms/step
 1/1 0s 128ms/step
 1/1 0s 128ms/step
 1/1 0s 129ms/step
 1/1 0s 130ms/step
 1/1 0s 129ms/step
 1/1 0s 130ms/step
 1/1 0s 128ms/step
 1/1 0s 127ms/step
 1/1 0s 128ms/step
 1/1 0s 126ms/step
 1/1 0s 128ms/step
 1/1 0s 133ms/step
 1/1 0s 128ms/step
 1/1 0s 129ms/step
 1/1 0s 128ms/step
 1/1 0s 127ms/step
 1/1 0s 128ms/step
 1/1 0s 128ms/step
 1/1 0s 127ms/step
 1/1 0s 128ms/step
 1/1 0s 128ms/step
 1/1 0s 127ms/step
 1/1 0s 128ms/step
 1/1 0s 128ms/step
 1/1 0s 127ms/step
 1/1 0s 128ms/step
 1/1 0s 128ms/step
 1/1 0s 127ms/step
 1/1 0s 128ms/step
 1/1 0s 128ms/step
 1/1 0s 127ms/step
 1/1 0s 128ms/step
 1/1 0s 128ms/step
 1/1 0s 127ms/step
 1/1 0s 128ms/step
 1/1 0s 128ms/step
 1/1 0s 131ms/step
 1/1 0s 128ms/step
 1/1 0s 127ms/step
 1/1 0s 128ms/step
 1/1 0s 128ms/step
 1/1 0s 127ms/step
 1/1 0s 127ms/step
 1/1 0s 130ms/step
 1/1 0s 129ms/step
 1/1 0s 128ms/step
 1/1 0s 128ms/step
 1/1 0s 133ms/step
 1/1 0s 63ms/step

2025-06-12 00:57:46.550179: I tensorflow/core/framework/local_rendezvous.cc:407] Local rendezvous is aborting with status: OUT_OF_RANGE: End of sequence
 <Figure size 1000x800 with 0 Axes>



O modelo demonstra bom desempenho geral, com elevado número de classificações corretas nas classes 'akiec', 'df', 'nv' e 'vasc'. Ainda assim, persistem confusões importantes entre as classes 'bkl', 'mel' e 'nv', o que é comum devido à semelhança visual entre essas lesões. Essas trocas podem impactar a performance clínica e sugerem a necessidade de mais dados ou técnicas para reforçar a distinção entre essas categorias.

Salvar o Modelo

```
In [ ]: model_t.save("modelt_3B_optuna_sem_data_aug_RMS_SGD.keras")
```