Modelo T com data augmentation e optimizer Adam

```
import json
import os
import matplotlib.pyplot as plt
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import GlobalAveragePooling2D, Dropout,
Dense, BatchNormalization
from tensorflow.keras.callbacks import ModelCheckpoint, EarlyStopping,
ReduceLROnPlateau, CSVLogger
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications import ResNet50
from tensorflow.keras.regularizers import l2
from tensorflow.keras.mixed_precision import set_global_policy
```

Definição de constantes

```
# MIX precision training -- facilita no treino!
set_global_policy('mixed_float16')

os.environ['TF_CPP_MIN_LOG_LEVEL'] = '2'

# CONSTANTES
BATCH_SIZE = 64
IMG_SIZE = 150
NUM_CLASSES = 10  # nº classes para identificar
NUM_EPOCHS = 60
LEARNING_RATE = 0.0001
DENSE_LAYERS = [1024, 512, 256, 128]
```

```
INFO:tensorflow:Mixed precision compatibility check (mixed_float16):
OK
Your GPU will likely run quickly with dtype policy mixed_float16 as it
has compute capability of at least 7.0. Your GPU: NVIDIA GeForce RTX
4070, compute capability 8.9
```

Data set

```
# Folders do dataset
train_dirs = ['./dataset/train/train1', './dataset/train/train2',
              './dataset/train/train3', './dataset/train/train5']
```

```
validation_dir = './dataset/validation'
test_dir = './dataset/test'
```

Data augmentation

Aqui decidimos aplicar um data augmentation mais robusto, até para testar novos parametros o que se mostrou efetivo, (mesmo que a diferença nao seja tão nítida) é possivel constatar uma evolução

```
# Data Augmentation
train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=20,  # Increase rotation range
    width_shift_range=0.1,
    height_shift_range=0.1,
    shear_range=0.1,
    zoom_range=0.1,
    horizontal_flip=True,
    vertical_flip=True,  # Adicionar flip vertical
    brightness_range=[0.6, 1.4],  # Adicionar range te brilho
    fill_mode='nearest')

validation_datagen = ImageDataGenerator(rescale=1./255)
test_datagen = ImageDataGenerator(rescale=1./255)

# training generators
train_generators = [train_datagen.flow_from_directory(
    train_dir,
    target_size=(IMG_SIZE, IMG_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='categorical') for train_dir in train_dirs]

# Necessário para juntar os trainning generators and repeat


def combined_generator(generators):
    while True:
        for generator in generators:
            for batch in generator:
                yield batch


train_generator = combined_generator(train_generators)

# Validation e test generators
validation_generator = validation_datagen.flow_from_directory(
    validation_dir,
    target_size=(IMG_SIZE, IMG_SIZE),
```

```
    batch_size=BATCH_SIZE,
    class_mode='categorical')

test_generator = test_datagen.flow_from_directory(
    test_dir,
    target_size=(IMG_SIZE, IMG_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='categorical')

# load do modelo ResNet50 - deixar include_top=False
base_model = ResNet50(weights='imagenet', include_top=False,
                      input_shape=(IMG_SIZE, IMG_SIZE, 3))
```

```
Found 10000 images belonging to 10 classes.
Found 10000 images belonging to 10 classes.
Found 10000 images belonging to 10 classes.
Found 10000 images belonging to 10 classes.
Found 10000 images belonging to 10 classes.
Found 10000 images belonging to 10 classes.
```

UnFreeze Layers - 100 layers

Aqui descongelamos as ultimas 100 camadas antes das 50 descongeladas no modelo T sem data augmentaion para comparar os resultados, e percebemos que a melhoria não foi assim tão significativa, dado que o tempo de treino é bastante superior quanto mais camadas descongeladas.

```
# Descongelar camadas (nao meter valores demasiado altos)
for layer in base_model.layers[-100:]:
    layer.trainable = True
```

Funções para obter as métricas

```
from tensorflow.keras import backend as K
from tensorflow.keras.metrics import Metric

class Precision(Metric):
    def __init__(self, name='precision', **kwargs):
        super(Precision, self).__init__(name=name, **kwargs)
        self.true_positives = self.add_weight(name='tp',
initializer='zeros')
        self.predicted_positives = self.add_weight(name='pp',
initializer='zeros')

    def update_state(self, y_true, y_pred, sample_weight=None):
        y_pred = K.round(y_pred)
```

```
        y_true = K.cast(y_true, 'float32')
        self.true_positives.assign_add(K.sum(y_true * y_pred))
        self.predicted_positives.assign_add(K.sum(y_pred))

    def result(self):
        return self.true_positives / (self.predicted_positives +
K.epsilon())

    def reset_states(self):
        self.true_positives.assign(0)
        self.predicted_positives.assign(0)

class Recall(Metric):
    def __init__(self, name='recall', **kwargs):
        super(Recall, self).__init__(name=name, **kwargs)
        self.true_positives = self.add_weight(name='tp',
initializer='zeros')
        self.actual_positives = self.add_weight(name='ap',
initializer='zeros')

    def update_state(self, y_true, y_pred, sample_weight=None):
        y_pred = K.round(y_pred)
        y_true = K.cast(y_true, 'float32')
        self.true_positives.assign_add(K.sum(y_true * y_pred))
        self.actual_positives.assign_add(K.sum(y_true))

    def result(self):
        return self.true_positives / (self.actual_positives +
K.epsilon())

    def reset_states(self):
        self.true_positives.assign(0)
        self.actual_positives.assign(0)

class F1Score(Metric):
    def __init__(self, name='f1_score', **kwargs):
        super(F1Score, self).__init__(name=name, **kwargs)
        self.precision = Precision()
        self.recall = Recall()

    def update_state(self, y_true, y_pred, sample_weight=None):
        self.precision.update_state(y_true, y_pred)
        self.recall.update_state(y_true, y_pred)

    def result(self):
        precision = self.precision.result()
        recall = self.recall.result()
        return 2 * ((precision * recall) / (precision + recall +
K.epsilon()))
```

```
    def reset_states(self):
        self.precision.reset_states()
        self.recall.reset_states()
```

Definir o modelo

```
# Definir as layers do modelo com parametros ajustados para reduzir o
overfitting
model = Sequential([
    base_model,
    BatchNormalization(),
    GlobalAveragePooling2D(),
    # Increase model complexity
    Dense(DENSE_LAYERS[0], activation='relu',
kernel_regularizer=l2(0.03)),
    Dropout(0.5),  # High dropout rate for regularization
    BatchNormalization(),
    Dense(DENSE_LAYERS[1], activation='relu',
kernel_regularizer=l2(0.03)),
    Dropout(0.5),
    BatchNormalization(),
    Dense(DENSE_LAYERS[2], activation='relu',
kernel_regularizer=l2(0.03)),
    Dropout(0.5),
    Dense(DENSE_LAYERS[3], activation='relu',
kernel_regularizer=l2(0.03)),
    Dropout(0.5),
    BatchNormalization(),
    Dense(NUM_CLASSES, activation='softmax', dtype='float32')
])

# Compilar o modelo
model.compile(optimizer=Adam(learning_rate=LEARNING_RATE),
              loss='categorical_crossentropy',
              metrics=['accuracy', Precision(), Recall(), F1Score()])

model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 resnet50 (Functional)       (None, 5, 5, 2048)        23587712

 batch_normalization (BatchN  (None, 5, 5, 2048)       8192
 ormalization)

 global_average_pooling2d (G  (None, 2048)             0
```

```
 lobalAveragePooling2D)

 dense (Dense)               (None, 1024)              2098176

 dropout (Dropout)           (None, 1024)              0

 batch_normalization_1 (Batc  (None, 1024)             4096
 hNormalization)

 dense_1 (Dense)             (None, 512)               524800

 dropout_1 (Dropout)         (None, 512)               0

 batch_normalization_2 (Batc  (None, 512)              2048
 hNormalization)

 dense_2 (Dense)             (None, 256)               131328

 dropout_2 (Dropout)         (None, 256)               0

 dense_3 (Dense)             (None, 128)               32896

 dropout_3 (Dropout)         (None, 128)               0

 batch_normalization_3 (Batc  (None, 128)              512
 hNormalization)

 dense_4 (Dense)             (None, 10)                1290

=================================================================
Total params: 26,391,050
Trainable params: 26,330,506
Non-trainable params: 60,544
_________________________________________________________________
```

Callbacks

```
# CALLBACKS
os.makedirs('logs', exist_ok=True)
checkpoint =
ModelCheckpoint(f'models/modelo_T_com_data_augmentation_adam.keras',
                                monitor='val_accuracy', verbose=1,
save_best_only=True, mode='max')
early_stopping = EarlyStopping(
    monitor='val_loss', patience=10, restore_best_weights=True)  #
Increased patience
reduce_lr = ReduceLROnPlateau(
    monitor='val_loss', factor=0.2, patience=4, min_lr=1e-7,
```

```
verbose=1)  # More aggressive schedule
csv_logger = CSVLogger(
    f'logs/modelo_T_com_data_augmentation_adam.csv', separator=',',
append=False)

# calcular passos por epoch
steps_per_epoch = sum([gen.samples // BATCH_SIZE for gen in
train_generators])
validation_steps = validation_generator.samples // BATCH_SIZE
```

Resultados

```
# calcular passos por epoch
# Treinar o modelo - Nao tirar os callbacks
history = model.fit(
    train_generator,
    steps_per_epoch=steps_per_epoch,
    epochs=NUM_EPOCHS,
    validation_data=validation_generator,
    validation_steps=validation_steps,
    callbacks=[checkpoint, early_stopping, reduce_lr, csv_logger]
)

# Avaliar o modelo no test generator
# Avaliar o modelo no test generator
results = model.evaluate(test_generator)
loss, accuracy, precision, recall, f1_score = results[:5]
print(f"Test Loss: {loss}")
print(f"Test Accuracy: {accuracy}")
print(f"Test Precision: {precision}")
print(f"Test Recall: {recall}")
print(f"Test F1 Score: {f1_score}")
```

```
Epoch 1/60
624/624 [==============================] - ETA: 0s - loss: 54.2261 -
accuracy: 0.1960 - precision: 0.3136 - recall: 0.0659 - f1_score:
0.1089

c:\Users\USER\.conda\envs\py310\lib\site-packages\keras\engine\
training.py:2319: UserWarning: Metric Precision implements a
`reset_states()` method; rename it to `reset_state()` (without the
final "s"). The name `reset_states()` has been deprecated to improve
API consistency.
  m.reset_state()
c:\Users\USER\.conda\envs\py310\lib\site-packages\keras\engine\
training.py:2319: UserWarning: Metric Recall implements a
`reset_states()` method; rename it to `reset_state()` (without the
final "s"). The name `reset_states()` has been deprecated to improve
```

```
API consistency.
  m.reset_state()
c:\Users\USER\.conda\envs\py310\lib\site-packages\keras\engine\
training.py:2319: UserWarning: Metric F1Score implements a
`reset_states()` method; rename it to `reset_state()` (without the
final "s"). The name `reset_states()` has been deprecated to improve
API consistency.
  m.reset_state()


Epoch 1: val_accuracy improved from -inf to 0.14233, saving model to
models\modelo_T_com_data_augmentation_adam_adam.keras
624/624 [==============================] - 185s 274ms/step - loss:
54.2261 - accuracy: 0.1960 - precision: 0.3136 - recall: 0.0659 -
f1_score: 0.1089 - val_loss: 34.5487 - val_accuracy: 0.1423 -
val_precision: 0.0000e+00 - val_recall: 0.0000e+00 - val_f1_score:
0.0000e+00 - lr: 1.0000e-04
Epoch 2/60
624/624 [==============================] - ETA: 0s - loss: 21.9455 -
accuracy: 0.4836 - precision: 0.6536 - recall: 0.2937 - f1_score:
0.4052
Epoch 2: val_accuracy improved from 0.14233 to 0.72516, saving model
to models\modelo_T_com_data_augmentation_adam_adam.keras
624/624 [==============================] - 167s 268ms/step - loss:
21.9455 - accuracy: 0.4836 - precision: 0.6536 - recall: 0.2937 -
f1_score: 0.4052 - val_loss: 12.4552 - val_accuracy: 0.7252 -
val_precision: 0.8539 - val_recall: 0.5697 - val_f1_score: 0.6834 -
lr: 1.0000e-04
Epoch 3/60
624/624 [==============================] - ETA: 0s - loss: 8.1400 -
accuracy: 0.6660 - precision: 0.7825 - recall: 0.5227 - f1_score:
0.6267
Epoch 3: val_accuracy improved from 0.72516 to 0.78746, saving model
to models\modelo_T_com_data_augmentation_adam_adam.keras
624/624 [==============================] - 167s 267ms/step - loss:
8.1400 - accuracy: 0.6660 - precision: 0.7825 - recall: 0.5227 -
f1_score: 0.6267 - val_loss: 4.6251 - val_accuracy: 0.7875 -
val_precision: 0.8772 - val_recall: 0.6931 - val_f1_score: 0.7744 -
lr: 1.0000e-04
Epoch 4/60
624/624 [==============================] - ETA: 0s - loss: 3.3969 -
accuracy: 0.7371 - precision: 0.8242 - recall: 0.6336 - f1_score:
0.7164
Epoch 4: val_accuracy improved from 0.78746 to 0.83313, saving model
to models\modelo_T_com_data_augmentation_adam_adam.keras
624/624 [==============================] - 166s 267ms/step - loss:
3.3969 - accuracy: 0.7371 - precision: 0.8242 - recall: 0.6336 -
f1_score: 0.7164 - val_loss: 2.0749 - val_accuracy: 0.8331 -
val_precision: 0.8833 - val_recall: 0.7769 - val_f1_score: 0.8267 -
lr: 1.0000e-04
```

```
Epoch 5/60
624/624 [==============================] - ETA: 0s - loss: 1.8381 -
accuracy: 0.7725 - precision: 0.8456 - recall: 0.6903 - f1_score:
0.7601
Epoch 5: val_accuracy did not improve from 0.83313
624/624 [==============================] - 165s 265ms/step - loss:
1.8381 - accuracy: 0.7725 - precision: 0.8456 - recall: 0.6903 -
f1_score: 0.7601 - val_loss: 1.3554 - val_accuracy: 0.8133 -
val_precision: 0.8647 - val_recall: 0.7640 - val_f1_score: 0.8112 -
lr: 1.0000e-04
Epoch 6/60
624/624 [==============================] - ETA: 0s - loss: 1.2571 -
accuracy: 0.7973 - precision: 0.8570 - recall: 0.7278 - f1_score:
0.7871
Epoch 6: val_accuracy did not improve from 0.83313
624/624 [==============================] - 166s 266ms/step - loss:
1.2571 - accuracy: 0.7973 - precision: 0.8570 - recall: 0.7278 -
f1_score: 0.7871 - val_loss: 1.0575 - val_accuracy: 0.8084 -
val_precision: 0.8514 - val_recall: 0.7657 - val_f1_score: 0.8063 -
lr: 1.0000e-04
Epoch 7/60
624/624 [==============================] - ETA: 0s - loss: 1.0065 -
accuracy: 0.8186 - precision: 0.8702 - recall: 0.7608 - f1_score:
0.8118
Epoch 7: val_accuracy did not improve from 0.83313
624/624 [==============================] - 165s 264ms/step - loss:
1.0065 - accuracy: 0.8186 - precision: 0.8702 - recall: 0.7608 -
f1_score: 0.8118 - val_loss: 1.0067 - val_accuracy: 0.7863 -
val_precision: 0.8286 - val_recall: 0.7479 - val_f1_score: 0.7862 -
lr: 1.0000e-04
Epoch 8/60
624/624 [==============================] - ETA: 0s - loss: 0.8948 -
accuracy: 0.8255 - precision: 0.8741 - recall: 0.7742 - f1_score:
0.8211
Epoch 8: val_accuracy did not improve from 0.83313
624/624 [==============================] - 165s 265ms/step - loss:
0.8948 - accuracy: 0.8255 - precision: 0.8741 - recall: 0.7742 -
f1_score: 0.8211 - val_loss: 0.8668 - val_accuracy: 0.8132 -
val_precision: 0.8417 - val_recall: 0.7926 - val_f1_score: 0.8164 -
lr: 1.0000e-04
Epoch 9/60
624/624 [==============================] - ETA: 0s - loss: 0.8092 -
accuracy: 0.8404 - precision: 0.8833 - recall: 0.7968 - f1_score:
0.8378
Epoch 9: val_accuracy did not improve from 0.83313
624/624 [==============================] - 165s 265ms/step - loss:
0.8092 - accuracy: 0.8404 - precision: 0.8833 - recall: 0.7968 -
f1_score: 0.8378 - val_loss: 0.8013 - val_accuracy: 0.8220 -
val_precision: 0.8499 - val_recall: 0.8028 - val_f1_score: 0.8257 -
```

```
lr: 1.0000e-04
Epoch 10/60
624/624 [==============================] - ETA: 0s - loss: 0.7587 -
accuracy: 0.8495 - precision: 0.8875 - recall: 0.8096 - f1_score:
0.8468
Epoch 10: val_accuracy did not improve from 0.83313
624/624 [==============================] - 165s 265ms/step - loss:
0.7587 - accuracy: 0.8495 - precision: 0.8875 - recall: 0.8096 -
f1_score: 0.8468 - val_loss: 0.7663 - val_accuracy: 0.8293 -
val_precision: 0.8624 - val_recall: 0.8054 - val_f1_score: 0.8329 -
lr: 1.0000e-04
Epoch 11/60
624/624 [==============================] - ETA: 0s - loss: 0.7058 -
accuracy: 0.8607 - precision: 0.8968 - recall: 0.8251 - f1_score:
0.8595
Epoch 11: val_accuracy improved from 0.83313 to 0.83494, saving model
to models\modelo_T_com_data_augmentation_adam_adam.keras
624/624 [==============================] - 166s 267ms/step - loss:
0.7058 - accuracy: 0.8607 - precision: 0.8968 - recall: 0.8251 -
f1_score: 0.8595 - val_loss: 0.7537 - val_accuracy: 0.8349 -
val_precision: 0.8562 - val_recall: 0.8195 - val_f1_score: 0.8375 -
lr: 1.0000e-04
Epoch 12/60
624/624 [==============================] - ETA: 0s - loss: 0.6781 -
accuracy: 0.8669 - precision: 0.9002 - recall: 0.8341 - f1_score:
0.8659
Epoch 12: val_accuracy did not improve from 0.83494
624/624 [==============================] - 166s 266ms/step - loss:
0.6781 - accuracy: 0.8669 - precision: 0.9002 - recall: 0.8341 -
f1_score: 0.8659 - val_loss: 0.7793 - val_accuracy: 0.8315 -
val_precision: 0.8599 - val_recall: 0.8132 - val_f1_score: 0.8359 -
lr: 1.0000e-04
Epoch 13/60
624/624 [==============================] - ETA: 0s - loss: 0.6483 -
accuracy: 0.8742 - precision: 0.9058 - recall: 0.8438 - f1_score:
0.8737
Epoch 13: val_accuracy did not improve from 0.83494
624/624 [==============================] - 166s 266ms/step - loss:
0.6483 - accuracy: 0.8742 - precision: 0.9058 - recall: 0.8438 -
f1_score: 0.8737 - val_loss: 0.9145 - val_accuracy: 0.8076 -
val_precision: 0.8289 - val_recall: 0.7952 - val_f1_score: 0.8117 -
lr: 1.0000e-04
Epoch 14/60
624/624 [==============================] - ETA: 0s - loss: 0.6276 -
accuracy: 0.8817 - precision: 0.9097 - recall: 0.8549 - f1_score:
0.8815
Epoch 14: val_accuracy did not improve from 0.83494
624/624 [==============================] - 166s 266ms/step - loss:
0.6276 - accuracy: 0.8817 - precision: 0.9097 - recall: 0.8549 -
```

```
f1_score: 0.8815 - val_loss: 0.7960 - val_accuracy: 0.8314 -
val_precision: 0.8526 - val_recall: 0.8154 - val_f1_score: 0.8336 -
lr: 1.0000e-04
Epoch 15/60
624/624 [==============================] - ETA: 0s - loss: 0.6171 -
accuracy: 0.8843 - precision: 0.9124 - recall: 0.8579 - f1_score:
0.8843
Epoch 15: val_accuracy did not improve from 0.83494

Epoch 15: ReduceLROnPlateau reducing learning rate to
1.9999999494757503e-05.
624/624 [==============================] - 165s 265ms/step - loss:
0.6171 - accuracy: 0.8843 - precision: 0.9124 - recall: 0.8579 -
f1_score: 0.8843 - val_loss: 0.8935 - val_accuracy: 0.8086 -
val_precision: 0.8332 - val_recall: 0.7936 - val_f1_score: 0.8129 -
lr: 1.0000e-04
Epoch 16/60
624/624 [==============================] - ETA: 0s - loss: 0.4505 -
accuracy: 0.9230 - precision: 0.9405 - recall: 0.9064 - f1_score:
0.9231
Epoch 16: val_accuracy improved from 0.83494 to 0.88011, saving model
to models\modelo_T_com_data_augmentation_adam_adam.keras
624/624 [==============================] - 167s 268ms/step - loss:
0.4505 - accuracy: 0.9230 - precision: 0.9405 - recall: 0.9064 -
f1_score: 0.9231 - val_loss: 0.5454 - val_accuracy: 0.8801 -
val_precision: 0.8921 - val_recall: 0.8699 - val_f1_score: 0.8808 -
lr: 2.0000e-05
Epoch 17/60
624/624 [==============================] - ETA: 0s - loss: 0.3569 -
accuracy: 0.9377 - precision: 0.9508 - recall: 0.9249 - f1_score:
0.9377
Epoch 17: val_accuracy improved from 0.88011 to 0.88752, saving model
to models\modelo_T_com_data_augmentation_adam_adam.keras
624/624 [==============================] - 167s 268ms/step - loss:
0.3569 - accuracy: 0.9377 - precision: 0.9508 - recall: 0.9249 -
f1_score: 0.9377 - val_loss: 0.5075 - val_accuracy: 0.8875 -
val_precision: 0.8985 - val_recall: 0.8791 - val_f1_score: 0.8887 -
lr: 2.0000e-05
Epoch 18/60
624/624 [==============================] - ETA: 0s - loss: 0.3205 -
accuracy: 0.9430 - precision: 0.9553 - recall: 0.9322 - f1_score:
0.9436
Epoch 18: val_accuracy did not improve from 0.88752
624/624 [==============================] - 165s 265ms/step - loss:
0.3205 - accuracy: 0.9430 - precision: 0.9553 - recall: 0.9322 -
f1_score: 0.9436 - val_loss: 0.5404 - val_accuracy: 0.8787 -
val_precision: 0.8885 - val_recall: 0.8714 - val_f1_score: 0.8799 -
lr: 2.0000e-05
Epoch 19/60
```

```
624/624 [==============================] - ETA: 0s - loss: 0.2858 -
accuracy: 0.9508 - precision: 0.9607 - recall: 0.9406 - f1_score:
0.9506
Epoch 19: val_accuracy did not improve from 0.88752
624/624 [==============================] - 165s 265ms/step - loss:
0.2858 - accuracy: 0.9508 - precision: 0.9607 - recall: 0.9406 -
f1_score: 0.9506 - val_loss: 0.5321 - val_accuracy: 0.8792 -
val_precision: 0.8897 - val_recall: 0.8736 - val_f1_score: 0.8816 -
lr: 2.0000e-05
Epoch 20/60
624/624 [==============================] - ETA: 0s - loss: 0.2715 -
accuracy: 0.9539 - precision: 0.9619 - recall: 0.9452 - f1_score:
0.9535
Epoch 20: val_accuracy did not improve from 0.88752
624/624 [==============================] - 166s 266ms/step - loss:
0.2715 - accuracy: 0.9539 - precision: 0.9619 - recall: 0.9452 -
f1_score: 0.9535 - val_loss: 0.5301 - val_accuracy: 0.8827 -
val_precision: 0.8912 - val_recall: 0.8767 - val_f1_score: 0.8839 -
lr: 2.0000e-05
Epoch 21/60
624/624 [==============================] - ETA: 0s - loss: 0.2632 -
accuracy: 0.9545 - precision: 0.9629 - recall: 0.9468 - f1_score:
0.9548
Epoch 21: val_accuracy did not improve from 0.88752

Epoch 21: ReduceLROnPlateau reducing learning rate to
3.999999898951501e-06.
624/624 [==============================] - 166s 266ms/step - loss:
0.2632 - accuracy: 0.9545 - precision: 0.9629 - recall: 0.9468 -
f1_score: 0.9548 - val_loss: 0.5169 - val_accuracy: 0.8831 -
val_precision: 0.8924 - val_recall: 0.8793 - val_f1_score: 0.8858 -
lr: 2.0000e-05
Epoch 22/60
624/624 [==============================] - ETA: 0s - loss: 0.2378 -
accuracy: 0.9600 - precision: 0.9679 - recall: 0.9532 - f1_score:
0.9605
Epoch 22: val_accuracy improved from 0.88752 to 0.89042, saving model
to models\modelo_T_com_data_augmentation_adam_adam.keras
624/624 [==============================] - 167s 267ms/step - loss:
0.2378 - accuracy: 0.9600 - precision: 0.9679 - recall: 0.9532 -
f1_score: 0.9605 - val_loss: 0.4888 - val_accuracy: 0.8904 -
val_precision: 0.8983 - val_recall: 0.8851 - val_f1_score: 0.8916 -
lr: 4.0000e-06
Epoch 23/60
624/624 [==============================] - ETA: 0s - loss: 0.2173 -
accuracy: 0.9635 - precision: 0.9705 - recall: 0.9570 - f1_score:
0.9637
Epoch 23: val_accuracy did not improve from 0.89042
624/624 [==============================] - 166s 266ms/step - loss:
```

```
0.2173 - accuracy: 0.9635 - precision: 0.9705 - recall: 0.9570 -
f1_score: 0.9637 - val_loss: 0.4903 - val_accuracy: 0.8894 -
val_precision: 0.8982 - val_recall: 0.8843 - val_f1_score: 0.8912 -
lr: 4.0000e-06
Epoch 24/60
624/624 [==============================] - ETA: 0s - loss: 0.2097 -
accuracy: 0.9657 - precision: 0.9720 - recall: 0.9598 - f1_score:
0.9659
Epoch 24: val_accuracy improved from 0.89042 to 0.89223, saving model
to models\modelo_T_com_data_augmentation_adam_adam.keras
624/624 [==============================] - 167s 268ms/step - loss:
0.2097 - accuracy: 0.9657 - precision: 0.9720 - recall: 0.9598 -
f1_score: 0.9659 - val_loss: 0.4815 - val_accuracy: 0.8922 -
val_precision: 0.8995 - val_recall: 0.8872 - val_f1_score: 0.8933 -
lr: 4.0000e-06
Epoch 25/60
624/624 [==============================] - ETA: 0s - loss: 0.2047 -
accuracy: 0.9665 - precision: 0.9732 - recall: 0.9605 - f1_score:
0.9668
Epoch 25: val_accuracy did not improve from 0.89223
624/624 [==============================] - 166s 266ms/step - loss:
0.2047 - accuracy: 0.9665 - precision: 0.9732 - recall: 0.9605 -
f1_score: 0.9668 - val_loss: 0.4787 - val_accuracy: 0.8919 -
val_precision: 0.8990 - val_recall: 0.8869 - val_f1_score: 0.8929 -
lr: 4.0000e-06
Epoch 26/60
624/624 [==============================] - ETA: 0s - loss: 0.1971 -
accuracy: 0.9676 - precision: 0.9731 - recall: 0.9626 - f1_score:
0.9679
Epoch 26: val_accuracy did not improve from 0.89223
624/624 [==============================] - 165s 265ms/step - loss:
0.1971 - accuracy: 0.9676 - precision: 0.9731 - recall: 0.9626 -
f1_score: 0.9679 - val_loss: 0.4849 - val_accuracy: 0.8901 -
val_precision: 0.8986 - val_recall: 0.8862 - val_f1_score: 0.8924 -
lr: 4.0000e-06
Epoch 27/60
624/624 [==============================] - ETA: 0s - loss: 0.1875 -
accuracy: 0.9707 - precision: 0.9759 - recall: 0.9654 - f1_score:
0.9706
Epoch 27: val_accuracy improved from 0.89223 to 0.89243, saving model
to models\modelo_T_com_data_augmentation_adam_adam.keras
624/624 [==============================] - 166s 267ms/step - loss:
0.1875 - accuracy: 0.9707 - precision: 0.9759 - recall: 0.9654 -
f1_score: 0.9706 - val_loss: 0.4841 - val_accuracy: 0.8924 -
val_precision: 0.9005 - val_recall: 0.8879 - val_f1_score: 0.8941 -
lr: 4.0000e-06
Epoch 28/60
624/624 [==============================] - ETA: 0s - loss: 0.1852 -
accuracy: 0.9702 - precision: 0.9747 - recall: 0.9650 - f1_score:
```

```
0.9698
Epoch 28: val_accuracy did not improve from 0.89243
624/624 [==============================] - 166s 266ms/step - loss:
0.1852 - accuracy: 0.9702 - precision: 0.9747 - recall: 0.9650 -
f1_score: 0.9698 - val_loss: 0.4751 - val_accuracy: 0.8918 -
val_precision: 0.8990 - val_recall: 0.8873 - val_f1_score: 0.8931 -
lr: 4.0000e-06
Epoch 29/60
624/624 [==============================] - ETA: 0s - loss: 0.1830 -
accuracy: 0.9694 - precision: 0.9749 - recall: 0.9647 - f1_score:
0.9698
Epoch 29: val_accuracy did not improve from 0.89243
624/624 [==============================] - 166s 266ms/step - loss:
0.1830 - accuracy: 0.9694 - precision: 0.9749 - recall: 0.9647 -
f1_score: 0.9698 - val_loss: 0.4811 - val_accuracy: 0.8914 -
val_precision: 0.8975 - val_recall: 0.8879 - val_f1_score: 0.8927 -
lr: 4.0000e-06
Epoch 30/60
624/624 [==============================] - ETA: 0s - loss: 0.1755 -
accuracy: 0.9722 - precision: 0.9768 - recall: 0.9672 - f1_score:
0.9720
Epoch 30: val_accuracy did not improve from 0.89243
624/624 [==============================] - 166s 266ms/step - loss:
0.1755 - accuracy: 0.9722 - precision: 0.9768 - recall: 0.9672 -
f1_score: 0.9720 - val_loss: 0.4828 - val_accuracy: 0.8895 -
val_precision: 0.8968 - val_recall: 0.8868 - val_f1_score: 0.8918 -
lr: 4.0000e-06
Epoch 31/60
624/624 [==============================] - ETA: 0s - loss: 0.1772 -
accuracy: 0.9705 - precision: 0.9752 - recall: 0.9654 - f1_score:
0.9702
Epoch 31: val_accuracy improved from 0.89243 to 0.89253, saving model
to models\modelo_T_com_data_augmentation_adam_adam.keras
624/624 [==============================] - 167s 267ms/step - loss:
0.1772 - accuracy: 0.9705 - precision: 0.9752 - recall: 0.9654 -
f1_score: 0.9702 - val_loss: 0.4794 - val_accuracy: 0.8925 -
val_precision: 0.8993 - val_recall: 0.8885 - val_f1_score: 0.8939 -
lr: 4.0000e-06
Epoch 32/60
624/624 [==============================] - ETA: 0s - loss: 0.1660 -
accuracy: 0.9729 - precision: 0.9781 - recall: 0.9691 - f1_score:
0.9736
Epoch 32: val_accuracy did not improve from 0.89253

Epoch 32: ReduceLROnPlateau reducing learning rate to
7.999999979801942e-07.
624/624 [==============================] - 167s 267ms/step - loss:
0.1660 - accuracy: 0.9729 - precision: 0.9781 - recall: 0.9691 -
f1_score: 0.9736 - val_loss: 0.4801 - val_accuracy: 0.8919 -
```

```
val_precision: 0.8979 - val_recall: 0.8876 - val_f1_score: 0.8927 -
lr: 4.0000e-06
Epoch 33/60
624/624 [==============================] - ETA: 0s - loss: 0.1676 -
accuracy: 0.9737 - precision: 0.9781 - recall: 0.9690 - f1_score:
0.9735
Epoch 33: val_accuracy improved from 0.89253 to 0.89263, saving model
to models\modelo_T_com_data_augmentation_adam_adam.keras
624/624 [==============================] - 167s 268ms/step - loss:
0.1676 - accuracy: 0.9737 - precision: 0.9781 - recall: 0.9690 -
f1_score: 0.9735 - val_loss: 0.4792 - val_accuracy: 0.8926 -
val_precision: 0.8996 - val_recall: 0.8888 - val_f1_score: 0.8942 -
lr: 8.0000e-07
Epoch 34/60
624/624 [==============================] - ETA: 0s - loss: 0.1619 -
accuracy: 0.9740 - precision: 0.9791 - recall: 0.9693 - f1_score:
0.9742
Epoch 34: val_accuracy did not improve from 0.89263
624/624 [==============================] - 166s 266ms/step - loss:
0.1619 - accuracy: 0.9740 - precision: 0.9791 - recall: 0.9693 -
f1_score: 0.9742 - val_loss: 0.4778 - val_accuracy: 0.8925 -
val_precision: 0.8992 - val_recall: 0.8886 - val_f1_score: 0.8939 -
lr: 8.0000e-07
Epoch 35/60
624/624 [==============================] - ETA: 0s - loss: 0.1602 -
accuracy: 0.9744 - precision: 0.9785 - recall: 0.9697 - f1_score:
0.9741
Epoch 35: val_accuracy improved from 0.89263 to 0.89343, saving model
to models\modelo_T_com_data_augmentation_adam_adam.keras
624/624 [==============================] - 167s 269ms/step - loss:
0.1602 - accuracy: 0.9744 - precision: 0.9785 - recall: 0.9697 -
f1_score: 0.9741 - val_loss: 0.4750 - val_accuracy: 0.8934 -
val_precision: 0.9009 - val_recall: 0.8889 - val_f1_score: 0.8949 -
lr: 8.0000e-07
Epoch 36/60
624/624 [==============================] - ETA: 0s - loss: 0.1597 -
accuracy: 0.9739 - precision: 0.9789 - recall: 0.9693 - f1_score:
0.9741
Epoch 36: val_accuracy did not improve from 0.89343

Epoch 36: ReduceLROnPlateau reducing learning rate to
1.6000000186977560e-07.
624/624 [==============================] - 166s 266ms/step - loss:
0.1597 - accuracy: 0.9739 - precision: 0.9789 - recall: 0.9693 -
f1_score: 0.9741 - val_loss: 0.4773 - val_accuracy: 0.8927 -
val_precision: 0.9003 - val_recall: 0.8883 - val_f1_score: 0.8943 -
lr: 8.0000e-07
Epoch 37/60
624/624 [==============================] - ETA: 0s - loss: 0.1597 -
```

```
accuracy: 0.9738 - precision: 0.9785 - recall: 0.9696 - f1_score:
0.9740
Epoch 37: val_accuracy improved from 0.89343 to 0.89363, saving model
to models\modelo_T_com_data_augmentation_adam_adam.keras
624/624 [==============================] - 167s 268ms/step - loss:
0.1597 - accuracy: 0.9738 - precision: 0.9785 - recall: 0.9696 -
f1_score: 0.9740 - val_loss: 0.4768 - val_accuracy: 0.8936 -
val_precision: 0.9007 - val_recall: 0.8896 - val_f1_score: 0.8951 -
lr: 1.6000e-07
Epoch 38/60
624/624 [==============================] - ETA: 0s - loss: 0.1607 -
accuracy: 0.9743 - precision: 0.9793 - recall: 0.9702 - f1_score:
0.9747
Epoch 38: val_accuracy did not improve from 0.89363
624/624 [==============================] - 167s 267ms/step - loss:
0.1607 - accuracy: 0.9743 - precision: 0.9793 - recall: 0.9702 -
f1_score: 0.9747 - val_loss: 0.4787 - val_accuracy: 0.8932 -
val_precision: 0.8994 - val_recall: 0.8888 - val_f1_score: 0.8941 -
lr: 1.6000e-07
Epoch 39/60
624/624 [==============================] - ETA: 0s - loss: 0.1600 -
accuracy: 0.9737 - precision: 0.9788 - recall: 0.9693 - f1_score:
0.9740
Epoch 39: val_accuracy improved from 0.89363 to 0.89413, saving model
to models\modelo_T_com_data_augmentation_adam_adam.keras
624/624 [==============================] - 170s 272ms/step - loss:
0.1600 - accuracy: 0.9737 - precision: 0.9788 - recall: 0.9693 -
f1_score: 0.9740 - val_loss: 0.4768 - val_accuracy: 0.8941 -
val_precision: 0.9013 - val_recall: 0.8899 - val_f1_score: 0.8956 -
lr: 1.6000e-07
Epoch 40/60
624/624 [==============================] - ETA: 0s - loss: 0.1599 -
accuracy: 0.9735 - precision: 0.9787 - recall: 0.9696 - f1_score:
0.9741
Epoch 40: val_accuracy did not improve from 0.89413

Epoch 40: ReduceLROnPlateau reducing learning rate to 1e-07.
624/624 [==============================] - 185s 296ms/step - loss:
0.1599 - accuracy: 0.9735 - precision: 0.9787 - recall: 0.9696 -
f1_score: 0.9741 - val_loss: 0.4751 - val_accuracy: 0.8941 -
val_precision: 0.9020 - val_recall: 0.8893 - val_f1_score: 0.8956 -
lr: 1.6000e-07
Epoch 41/60
624/624 [==============================] - ETA: 0s - loss: 0.1568 -
accuracy: 0.9758 - precision: 0.9797 - recall: 0.9714 - f1_score:
0.9755
Epoch 41: val_accuracy did not improve from 0.89413
624/624 [==============================] - 168s 270ms/step - loss:
0.1568 - accuracy: 0.9758 - precision: 0.9797 - recall: 0.9714 -
```

```
f1_score: 0.9755 - val_loss: 0.4768 - val_accuracy: 0.8933 -
val_precision: 0.9006 - val_recall: 0.8890 - val_f1_score: 0.8948 -
lr: 1.0000e-07
Epoch 42/60
624/624 [==============================] - ETA: 0s - loss: 0.1574 -
accuracy: 0.9746 - precision: 0.9796 - recall: 0.9705 - f1_score:
0.9750
Epoch 42: val_accuracy did not improve from 0.89413
624/624 [==============================] - 168s 269ms/step - loss:
0.1574 - accuracy: 0.9746 - precision: 0.9796 - recall: 0.9705 -
f1_score: 0.9750 - val_loss: 0.4780 - val_accuracy: 0.8934 -
val_precision: 0.9005 - val_recall: 0.8886 - val_f1_score: 0.8945 -
lr: 1.0000e-07
Epoch 43/60
624/624 [==============================] - ETA: 0s - loss: 0.1618 -
accuracy: 0.9732 - precision: 0.9778 - recall: 0.9690 - f1_score:
0.9734
Epoch 43: val_accuracy did not improve from 0.89413
624/624 [==============================] - 168s 270ms/step - loss:
0.1618 - accuracy: 0.9732 - precision: 0.9778 - recall: 0.9690 -
f1_score: 0.9734 - val_loss: 0.4767 - val_accuracy: 0.8931 -
val_precision: 0.9006 - val_recall: 0.8893 - val_f1_score: 0.8949 -
lr: 1.0000e-07
Epoch 44/60
624/624 [==============================] - ETA: 0s - loss: 0.1562 -
accuracy: 0.9750 - precision: 0.9797 - recall: 0.9711 - f1_score:
0.9754
Epoch 44: val_accuracy did not improve from 0.89413
624/624 [==============================] - 169s 271ms/step - loss:
0.1562 - accuracy: 0.9750 - precision: 0.9797 - recall: 0.9711 -
f1_score: 0.9754 - val_loss: 0.4758 - val_accuracy: 0.8928 -
val_precision: 0.9003 - val_recall: 0.8893 - val_f1_score: 0.8948 -
lr: 1.0000e-07
Epoch 45/60
624/624 [==============================] - ETA: 0s - loss: 0.1583 -
accuracy: 0.9753 - precision: 0.9795 - recall: 0.9713 - f1_score:
0.9754
Epoch 45: val_accuracy did not improve from 0.89413
624/624 [==============================] - 175s 281ms/step - loss:
0.1583 - accuracy: 0.9753 - precision: 0.9795 - recall: 0.9713 -
f1_score: 0.9754 - val_loss: 0.4748 - val_accuracy: 0.8940 -
val_precision: 0.9006 - val_recall: 0.8895 - val_f1_score: 0.8950 -
lr: 1.0000e-07
Epoch 46/60
624/624 [==============================] - ETA: 0s - loss: 0.1568 -
accuracy: 0.9752 - precision: 0.9800 - recall: 0.9714 - f1_score:
0.9757
Epoch 46: val_accuracy did not improve from 0.89413
624/624 [==============================] - 171s 274ms/step - loss:
```

```
0.1568 - accuracy: 0.9752 - precision: 0.9800 - recall: 0.9714 -
f1_score: 0.9757 - val_loss: 0.4760 - val_accuracy: 0.8932 -
val_precision: 0.9005 - val_recall: 0.8892 - val_f1_score: 0.8948 -
lr: 1.0000e-07
Epoch 47/60
624/624 [==============================] - ETA: 0s - loss: 0.1570 -
accuracy: 0.9748 - precision: 0.9789 - recall: 0.9708 - f1_score:
0.9748
Epoch 47: val_accuracy did not improve from 0.89413
624/624 [==============================] - 168s 269ms/step - loss:
0.1570 - accuracy: 0.9748 - precision: 0.9789 - recall: 0.9708 -
f1_score: 0.9748 - val_loss: 0.4755 - val_accuracy: 0.8935 -
val_precision: 0.9010 - val_recall: 0.8901 - val_f1_score: 0.8955 -
lr: 1.0000e-07
Epoch 48/60
624/624 [==============================] - ETA: 0s - loss: 0.1577 -
accuracy: 0.9750 - precision: 0.9790 - recall: 0.9708 - f1_score:
0.9749
Epoch 48: val_accuracy did not improve from 0.89413
624/624 [==============================] - 170s 273ms/step - loss:
0.1577 - accuracy: 0.9750 - precision: 0.9790 - recall: 0.9708 -
f1_score: 0.9749 - val_loss: 0.4749 - val_accuracy: 0.8939 -
val_precision: 0.9016 - val_recall: 0.8902 - val_f1_score: 0.8959 -
lr: 1.0000e-07
Epoch 49/60
624/624 [==============================] - ETA: 0s - loss: 0.1542 -
accuracy: 0.9758 - precision: 0.9797 - recall: 0.9717 - f1_score:
0.9757
Epoch 49: val_accuracy did not improve from 0.89413
624/624 [==============================] - 171s 274ms/step - loss:
0.1542 - accuracy: 0.9758 - precision: 0.9797 - recall: 0.9717 -
f1_score: 0.9757 - val_loss: 0.4748 - val_accuracy: 0.8937 -
val_precision: 0.9008 - val_recall: 0.8898 - val_f1_score: 0.8953 -
lr: 1.0000e-07
Epoch 50/60
624/624 [==============================] - ETA: 0s - loss: 0.1514 -
accuracy: 0.9763 - precision: 0.9812 - recall: 0.9726 - f1_score:
0.9769
Epoch 50: val_accuracy did not improve from 0.89413
624/624 [==============================] - 170s 273ms/step - loss:
0.1514 - accuracy: 0.9763 - precision: 0.9812 - recall: 0.9726 -
f1_score: 0.9769 - val_loss: 0.4754 - val_accuracy: 0.8935 -
val_precision: 0.9005 - val_recall: 0.8900 - val_f1_score: 0.8952 -
lr: 1.0000e-07
Epoch 51/60
624/624 [==============================] - ETA: 0s - loss: 0.1570 -
accuracy: 0.9754 - precision: 0.9797 - recall: 0.9713 - f1_score:
0.9755
Epoch 51: val_accuracy did not improve from 0.89413
624/624 [==============================] - 169s 272ms/step - loss:
```

```
0.1570 - accuracy: 0.9754 - precision: 0.9797 - recall: 0.9713 -
f1_score: 0.9755 - val_loss: 0.4765 - val_accuracy: 0.8931 -
val_precision: 0.9017 - val_recall: 0.8899 - val_f1_score: 0.8958 -
lr: 1.0000e-07
Epoch 52/60
624/624 [==============================] - ETA: 0s - loss: 0.1580 -
accuracy: 0.9742 - precision: 0.9794 - recall: 0.9705 - f1_score:
0.9749
Epoch 52: val_accuracy did not improve from 0.89413
624/624 [==============================] - 170s 273ms/step - loss:
0.1580 - accuracy: 0.9742 - precision: 0.9794 - recall: 0.9705 -
f1_score: 0.9749 - val_loss: 0.4755 - val_accuracy: 0.8931 -
val_precision: 0.9009 - val_recall: 0.8894 - val_f1_score: 0.8951 -
lr: 1.0000e-07
Epoch 53/60
624/624 [==============================] - ETA: 0s - loss: 0.1557 -
accuracy: 0.9756 - precision: 0.9799 - recall: 0.9714 - f1_score:
0.9756
Epoch 53: val_accuracy did not improve from 0.89413
624/624 [==============================] - 170s 273ms/step - loss:
0.1557 - accuracy: 0.9756 - precision: 0.9799 - recall: 0.9714 -
f1_score: 0.9756 - val_loss: 0.4755 - val_accuracy: 0.8934 -
val_precision: 0.9013 - val_recall: 0.8888 - val_f1_score: 0.8950 -
lr: 1.0000e-07
Epoch 54/60
624/624 [==============================] - ETA: 0s - loss: 0.1584 -
accuracy: 0.9748 - precision: 0.9789 - recall: 0.9709 - f1_score:
0.9749
Epoch 54: val_accuracy did not improve from 0.89413
624/624 [==============================] - 171s 275ms/step - loss:
0.1584 - accuracy: 0.9748 - precision: 0.9789 - recall: 0.9709 -
f1_score: 0.9749 - val_loss: 0.4764 - val_accuracy: 0.8927 -
val_precision: 0.9005 - val_recall: 0.8892 - val_f1_score: 0.8948 -
lr: 1.0000e-07
Epoch 55/60
624/624 [==============================] - ETA: 0s - loss: 0.1505 -
accuracy: 0.9766 - precision: 0.9803 - recall: 0.9726 - f1_score:
0.9764
Epoch 55: val_accuracy did not improve from 0.89413
624/624 [==============================] - 170s 272ms/step - loss:
0.1505 - accuracy: 0.9766 - precision: 0.9803 - recall: 0.9726 -
f1_score: 0.9764 - val_loss: 0.4749 - val_accuracy: 0.8939 -
val_precision: 0.9013 - val_recall: 0.8895 - val_f1_score: 0.8954 -
lr: 1.0000e-07
Epoch 56/60
624/624 [==============================] - ETA: 0s - loss: 0.1561 -
accuracy: 0.9746 - precision: 0.9793 - recall: 0.9707 - f1_score:
0.9750
Epoch 56: val_accuracy improved from 0.89413 to 0.89433, saving model
```

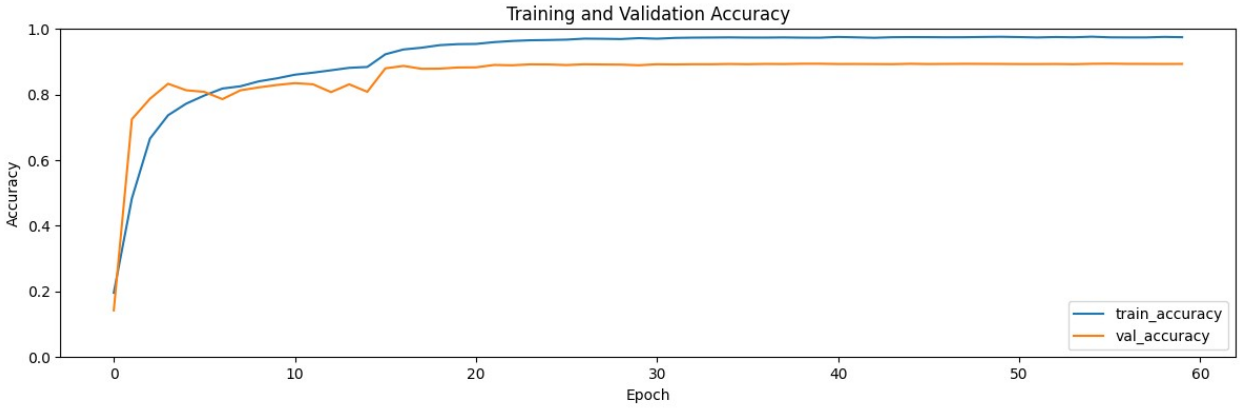
```
to models\modelo_T_com_data_augmentation_adam_adam.keras
624/624 [==============================] - 170s 273ms/step - loss:
0.1561 - accuracy: 0.9746 - precision: 0.9793 - recall: 0.9707 -
f1_score: 0.9750 - val_loss: 0.4727 - val_accuracy: 0.8943 -
val_precision: 0.9018 - val_recall: 0.8899 - val_f1_score: 0.8958 -
lr: 1.0000e-07
Epoch 57/60
624/624 [==============================] - ETA: 0s - loss: 0.1577 -
accuracy: 0.9743 - precision: 0.9791 - recall: 0.9703 - f1_score:
0.9747
Epoch 57: val_accuracy did not improve from 0.89433
624/624 [==============================] - 171s 274ms/step - loss:
0.1577 - accuracy: 0.9743 - precision: 0.9791 - recall: 0.9703 -
f1_score: 0.9747 - val_loss: 0.4755 - val_accuracy: 0.8937 -
val_precision: 0.9010 - val_recall: 0.8892 - val_f1_score: 0.8950 -
lr: 1.0000e-07
Epoch 58/60
624/624 [==============================] - ETA: 0s - loss: 0.1569 -
accuracy: 0.9743 - precision: 0.9791 - recall: 0.9706 - f1_score:
0.9748
Epoch 58: val_accuracy did not improve from 0.89433
624/624 [==============================] - 173s 278ms/step - loss:
0.1569 - accuracy: 0.9743 - precision: 0.9791 - recall: 0.9706 -
f1_score: 0.9748 - val_loss: 0.4752 - val_accuracy: 0.8936 -
val_precision: 0.9008 - val_recall: 0.8894 - val_f1_score: 0.8951 -
lr: 1.0000e-07
Epoch 59/60
624/624 [==============================] - ETA: 0s - loss: 0.1531 -
accuracy: 0.9760 - precision: 0.9805 - recall: 0.9713 - f1_score:
0.9759
Epoch 59: val_accuracy did not improve from 0.89433
624/624 [==============================] - 177s 284ms/step - loss:
0.1531 - accuracy: 0.9760 - precision: 0.9805 - recall: 0.9713 -
f1_score: 0.9759 - val_loss: 0.4768 - val_accuracy: 0.8934 -
val_precision: 0.9008 - val_recall: 0.8885 - val_f1_score: 0.8946 -
lr: 1.0000e-07
Epoch 60/60
624/624 [==============================] - ETA: 0s - loss: 0.1578 -
accuracy: 0.9750 - precision: 0.9792 - recall: 0.9710 - f1_score:
0.9751
Epoch 60: val_accuracy did not improve from 0.89433
624/624 [==============================] - 170s 272ms/step - loss:
0.1578 - accuracy: 0.9750 - precision: 0.9792 - recall: 0.9710 -
f1_score: 0.9751 - val_loss: 0.4760 - val_accuracy: 0.8936 -
val_precision: 0.9016 - val_recall: 0.8888 - val_f1_score: 0.8951 -
lr: 1.0000e-07
157/157 [==============================] - 6s 37ms/step - loss: 0.4905
- accuracy: 0.8898 - precision: 0.8963 - recall: 0.8860 - f1_score:
0.8911
```

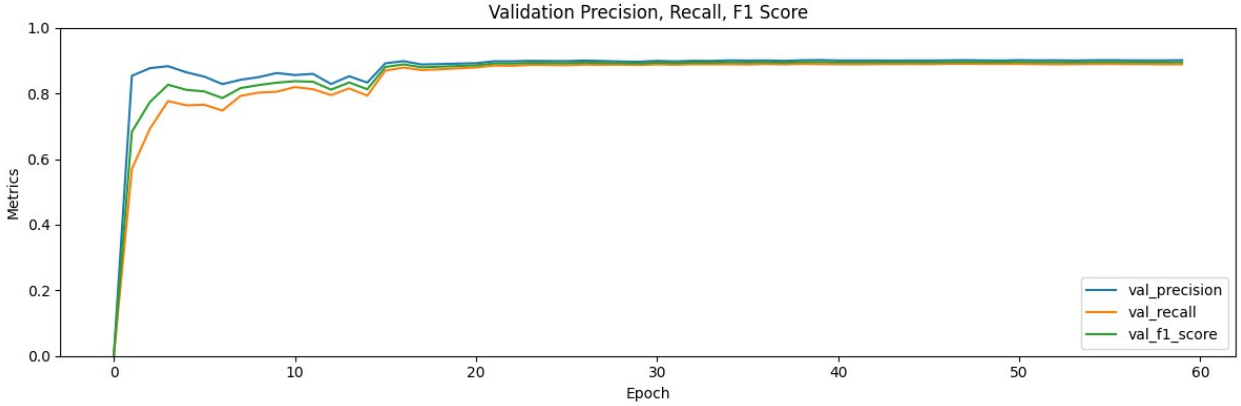
```
Test Loss: 0.4904648959636688
Test Accuracy: 0.8898000121116638
Test Precision: 0.8963075280189514
Test Recall: 0.8859999775886536
Test F1 Score: 0.89112389087677
```

```
plt.figure(figsize=(12, 8))
plt.subplot(2, 1, 1)
plt.plot(history.history['accuracy'], label='train_accuracy')
plt.plot(history.history['val_accuracy'], label='val_accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.ylim([0, 1])
plt.legend(loc='lower right')
plt.title('Training and Validation Accuracy')

plt.subplot(2, 1, 2)
plt.plot(history.history['val_precision'], label='val_precision')
plt.plot(history.history['val_recall'], label='val_recall')
plt.plot(history.history['val_f1_score'], label='val_f1_score')
plt.xlabel('Epoch')
plt.ylabel('Metrics')
plt.ylim([0, 1])
plt.legend(loc='lower right')
plt.title('Validation Precision, Recall, F1 Score')

plt.savefig(f'./plots/modelo_T_com_data_augmentation_adam.png')
plt.tight_layout()
#plt.show()
```


Training and Validation Accuracy
Accuracy
Epoch
train_accuracy
val_accuracy


Validation Precision, Recall, F1 Score
Metrics
Epoch
val_precision
val_recall
val_f1_score