

Modelo S com Data Augmentation

```
from tensorflow.keras.metrics import Metric
from tensorflow.keras import backend as K
import numpy as np
import tensorflow as tf
import matplotlib.pyplot as plt
import os
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Activation, Dropout, Flatten,
Dense, Conv2D, MaxPooling2D, BatchNormalization
from tensorflow.keras.callbacks import ModelCheckpoint, EarlyStopping,
CSVLogger, ReduceLROnPlateau
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.preprocessing.image import ImageDataGenerator

os.environ['TF_CPP_MIN_LOG_LEVEL'] = '2'
# CONSTANTES
BATCH_SIZE = 64
IMG_SIZE = 32
NUM_CLASSES = 10 # nº classes para identificar
NUM_EPOCHS = 100
LEARNING_RATE = 0.001
DENSE_LAYERS = [256, 512, 1024, 1024]

# 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

Testámos vários tipos de data augmentation, com mais e menos intensidade. Concluímos que os valores escolhidos (rotações de 15 graus, deslocamentos horizontais e verticais de 0,01 e horizontal_flip) proporcionavam os melhores resultados, mesmo que essa melhora seja tão pouco significativa que nos deixe a pensar se efetivamente compensa de todo este tempo extra de treino.

Chegamos à conclusão que dado que as imagens do dataset têm baixa resolução, adicionar demasiada data augmentation prejudicava os resultados. Isto ocorre porque transformações excessivas em imagens de baixa qualidade podem distorcer os dados de forma significativa, dificultando a capacidade do modelo de aprender características relevantes. Portanto, um equilíbrio cuidadoso foi necessário para melhorar a robustez do modelo sem comprometer a integridade dos dados visuais.

```

train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=15,
    width_shift_range=0.01,
    height_shift_range=0.01,
    horizontal_flip=True,
    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 junstar os trainning generators

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

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')

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.

```

Teste com outras intensidades de data augmentation:

```
train_datagen = ImageDataGenerator( rescale=1./255, rotation_range=15,  
width_shift_range=0.01, height_shift_range=0.01, shear_range=0.01, zoom_range=0.01,  
horizontal_flip=True, fill_mode='nearest')
```

Validation Accuracy Obtida: 0.8462539911270142

```
train_datagen = ImageDataGenerator( rescale=1./255, rotation_range=5,  
height_shift_range=0.01, horizontal_flip=True, fill_mode='nearest')
```

Validation Accuracy Obtida: 0.8409455418586731

```
train_datagen = ImageDataGenerator( rescale=1./255, rotation_range=2,  
width_shift_range=0.01, horizontal_flip=True, fill_mode='nearest')
```

Validation Accuracy Obtida: 0.8501634340286255

```
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()

model = Sequential([
    Conv2D(DENSE_LAYERS[0], (3, 3), input_shape=(IMG_SIZE, IMG_SIZE,
3)),
    BatchNormalization(),
    Activation('relu'),
    MaxPooling2D((2, 2)),
    Dropout(0.3),

    Conv2D(DENSE_LAYERS[1], (3, 3)),
    BatchNormalization(),
    Activation('relu'),
    MaxPooling2D((2, 2)),
    Dropout(0.5),

```

```

        Conv2D(DENSE_LAYERS[2], (3, 3)),
        BatchNormalization(),
        Activation('relu'),
        MaxPooling2D((2, 2)),
        Dropout(0.5),

        Flatten(),
        Dense(DENSE_LAYERS[3]),
        BatchNormalization(),
        Activation('relu'),
        Dropout(0.5),

        Dense(NUM_CLASSES, activation='softmax')
    ])

# 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 #
conv2d (Conv2D)	(None, 30, 30, 256)	7168
batch_normalization (Batch Normalization)	(None, 30, 30, 256)	1024
activation (Activation)	(None, 30, 30, 256)	0
max_pooling2d (MaxPooling2D)	(None, 15, 15, 256)	0
dropout (Dropout)	(None, 15, 15, 256)	0
conv2d_1 (Conv2D)	(None, 13, 13, 512)	1180160
batch_normalization_1 (Batch Normalization)	(None, 13, 13, 512)	2048
activation_1 (Activation)	(None, 13, 13, 512)	0
max_pooling2d_1 (MaxPooling2D)	(None, 6, 6, 512)	0
dropout_1 (Dropout)	(None, 6, 6, 512)	0

conv2d_2 (Conv2D)	(None, 4, 4, 1024)	4719616
batch_normalization_2 (Batch Normalization)	(None, 4, 4, 1024)	4096
activation_2 (Activation)	(None, 4, 4, 1024)	0
max_pooling2d_2 (MaxPooling2D)	(None, 2, 2, 1024)	0
dropout_2 (Dropout)	(None, 2, 2, 1024)	0
flatten (Flatten)	(None, 4096)	0
dense (Dense)	(None, 1024)	4195328
batch_normalization_3 (Batch Normalization)	(None, 1024)	4096
activation_3 (Activation)	(None, 1024)	0
dropout_3 (Dropout)	(None, 1024)	0
dense_1 (Dense)	(None, 10)	10250
=====		
Total params: 10,123,786		
Trainable params: 10,118,154		
Non-trainable params: 5,632		

Definir os Callbacks

Para salvar o melhor modelo com base na acurácia de validação

```
checkpoint = ModelCheckpoint(
    "models/modelo_S_com_data_augmentation_adam.keras",
    monitor='val_accuracy', verbose=1, save_best_only=True, mode='max')
```

Parar o treinamento se não houver melhoria na loss após x epochs

```
early_stopping = EarlyStopping(
    monitor='val_loss', patience=10, restore_best_weights=True)
```

Salvar para csv

```
csv_logger = CSVLogger(
    f'logs/modelo_S_com_data_augmentation_adam.csv', append=True)
```

Reduzir a learning rate se não houver melhoria na loss após x epochs (lembrar de deixar este valor sempre menor que a patience no early_stopping!!)

```

reduce_lr = ReduceLRonPlateau(
    monitor='val_loss', factor=0.5, patience=3, verbose=1)

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

# 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_generator.samples // BATCH_SIZE,
    callbacks=[checkpoint, early_stopping, csv_logger, reduce_lr]
)

# 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 1/100
624/624 [=====] - ETA: 0s - loss: 1.6637 -
accuracy: 0.4272 - precision: 0.5460 - recall: 0.2850 - f1_score:
0.3745

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 0.51332, saving model to models\modelo_S_com_data_augmentation_adam.keras

624/624 [=====] - 64s 86ms/step - loss: 1.6637 - accuracy: 0.4272 - precision: 0.5460 - recall: 0.2850 - f1_score: 0.3745 - val_loss: 1.3844 - val_accuracy: 0.5133 - val_precision: 0.7241 - val_recall: 0.3057 - val_f1_score: 0.4299 - lr: 0.0010

Epoch 2/100

624/624 [=====] - ETA: 0s - loss: 1.2339 - accuracy: 0.5618 - precision: 0.6893 - recall: 0.4292 - f1_score: 0.5290

Epoch 2: val_accuracy improved from 0.51332 to 0.58664, saving model to models\modelo_S_com_data_augmentation_adam.keras

624/624 [=====] - 41s 65ms/step - loss: 1.2339 - accuracy: 0.5618 - precision: 0.6893 - recall: 0.4292 - f1_score: 0.5290 - val_loss: 1.1930 - val_accuracy: 0.5866 - val_precision: 0.6948 - val_recall: 0.4849 - val_f1_score: 0.5712 - lr: 0.0010

Epoch 3/100

623/624 [=====>.] - ETA: 0s - loss: 1.0891 - accuracy: 0.6115 - precision: 0.7350 - recall: 0.4941 - f1_score: 0.5909

Epoch 3: val_accuracy did not improve from 0.58664

624/624 [=====] - 33s 53ms/step - loss: 1.0889 - accuracy: 0.6115 - precision: 0.7350 - recall: 0.4941 - f1_score: 0.5909 - val_loss: 1.6433 - val_accuracy: 0.4916 - val_precision: 0.5664 - val_recall: 0.4251 - val_f1_score: 0.4857 - lr: 0.0010

Epoch 4/100

623/624 [=====>.] - ETA: 0s - loss: 1.0074 - accuracy: 0.6451 - precision: 0.7558 - recall: 0.5382 - f1_score: 0.6287

Epoch 4: val_accuracy did not improve from 0.58664

624/624 [=====] - 33s 53ms/step - loss: 1.0073 - accuracy: 0.6451 - precision: 0.7560 - recall: 0.5381 - f1_score: 0.6287 - val_loss: 1.2640 - val_accuracy: 0.5733 - val_precision: 0.6641 - val_recall: 0.4999 - val_f1_score: 0.5704 - lr: 0.0010

Epoch 5/100

624/624 [=====] - ETA: 0s - loss: 0.9421 - accuracy: 0.6707 - precision: 0.7703 - recall: 0.5698 - f1_score: 0.6551

Epoch 5: val_accuracy improved from 0.58664 to 0.68830, saving model to models\modelo_S_com_data_augmentation_adam.keras

624/624 [=====] - 33s 53ms/step - loss: 0.9421 - accuracy: 0.6707 - precision: 0.7703 - recall: 0.5698 - f1_score: 0.6551 - val_loss: 0.9065 - val_accuracy: 0.6883 - val_precision: 0.8154 - val_recall: 0.5613 - val_f1_score: 0.6649 - lr: 0.0010


```
Epoch 6/100
624/624 [=====] - ETA: 0s - loss: 0.9013 -
accuracy: 0.6825 - precision: 0.7783 - recall: 0.5902 - f1_score:
0.6713
Epoch 6: val_accuracy did not improve from 0.68830
624/624 [=====] - 33s 53ms/step - loss:
0.9013 - accuracy: 0.6825 - precision: 0.7783 - recall: 0.5902 -
f1_score: 0.6713 - val_loss: 0.9239 - val_accuracy: 0.6822 -
val_precision: 0.7608 - val_recall: 0.6066 - val_f1_score: 0.6750 -
lr: 0.0010
Epoch 7/100
624/624 [=====] - ETA: 0s - loss: 0.8488 -
accuracy: 0.7023 - precision: 0.7922 - recall: 0.6177 - f1_score:
0.6941
Epoch 7: val_accuracy improved from 0.68830 to 0.70473, saving model
to models\modelo_S_com_data_augmentation_adam.keras
624/624 [=====] - 32s 52ms/step - loss:
0.8488 - accuracy: 0.7023 - precision: 0.7922 - recall: 0.6177 -
f1_score: 0.6941 - val_loss: 0.8522 - val_accuracy: 0.7047 -
val_precision: 0.7846 - val_recall: 0.6320 - val_f1_score: 0.7001 -
lr: 0.0010
Epoch 8/100
623/624 [=====>.] - ETA: 0s - loss: 0.8126 -
accuracy: 0.7180 - precision: 0.8038 - recall: 0.6343 - f1_score:
0.7091
Epoch 8: val_accuracy did not improve from 0.70473
624/624 [=====] - 32s 52ms/step - loss:
0.8124 - accuracy: 0.7181 - precision: 0.8039 - recall: 0.6344 -
f1_score: 0.7091 - val_loss: 1.0332 - val_accuracy: 0.6625 -
val_precision: 0.7245 - val_recall: 0.6025 - val_f1_score: 0.6579 -
lr: 0.0010
Epoch 9/100
624/624 [=====] - ETA: 0s - loss: 0.7815 -
accuracy: 0.7259 - precision: 0.8093 - recall: 0.6476 - f1_score:
0.7195
Epoch 9: val_accuracy did not improve from 0.70473
624/624 [=====] - 32s 52ms/step - loss:
0.7815 - accuracy: 0.7259 - precision: 0.8093 - recall: 0.6476 -
f1_score: 0.7195 - val_loss: 1.0233 - val_accuracy: 0.6643 -
val_precision: 0.7378 - val_recall: 0.5998 - val_f1_score: 0.6617 -
lr: 0.0010
Epoch 10/100
623/624 [=====>.] - ETA: 0s - loss: 0.7598 -
accuracy: 0.7350 - precision: 0.8153 - recall: 0.6592 - f1_score:
0.7290
Epoch 10: val_accuracy improved from 0.70473 to 0.70873, saving model
to models\modelo_S_com_data_augmentation_adam.keras
624/624 [=====] - 32s 52ms/step - loss:
0.7596 - accuracy: 0.7351 - precision: 0.8153 - recall: 0.6593 -
f1_score: 0.7291 - val_loss: 0.8323 - val_accuracy: 0.7087 -
```

```
val_precision: 0.7951 - val_recall: 0.6423 - val_f1_score: 0.7106 -  
lr: 0.0010  
Epoch 11/100  
623/624 [=====>.] - ETA: 0s - loss: 0.7315 -  
accuracy: 0.7440 - precision: 0.8179 - recall: 0.6731 - f1_score:  
0.7385  
Epoch 11: val_accuracy did not improve from 0.70873  
624/624 [=====] - 32s 52ms/step - loss:  
0.7322 - accuracy: 0.7437 - precision: 0.8177 - recall: 0.6730 -  
f1_score: 0.7383 - val_loss: 0.9851 - val_accuracy: 0.6774 -  
val_precision: 0.7392 - val_recall: 0.6284 - val_f1_score: 0.6793 -  
lr: 0.0010  
Epoch 12/100  
623/624 [=====>.] - ETA: 0s - loss: 0.7134 -  
accuracy: 0.7523 - precision: 0.8231 - recall: 0.6834 - f1_score:  
0.7468  
Epoch 12: val_accuracy did not improve from 0.70873  
624/624 [=====] - 32s 52ms/step - loss:  
0.7132 - accuracy: 0.7524 - precision: 0.8231 - recall: 0.6835 -  
f1_score: 0.7469 - val_loss: 1.0373 - val_accuracy: 0.6635 -  
val_precision: 0.7242 - val_recall: 0.6150 - val_f1_score: 0.6651 -  
lr: 0.0010  
Epoch 13/100  
624/624 [=====] - ETA: 0s - loss: 0.6828 -  
accuracy: 0.7599 - precision: 0.8302 - recall: 0.6942 - f1_score:  
0.7561  
Epoch 13: val_accuracy did not improve from 0.70873  
  
Epoch 13: ReduceLROnPlateau reducing learning rate to  
0.00050000000237487257.  
624/624 [=====] - 33s 52ms/step - loss:  
0.6828 - accuracy: 0.7599 - precision: 0.8302 - recall: 0.6942 -  
f1_score: 0.7561 - val_loss: 1.0965 - val_accuracy: 0.6664 -  
val_precision: 0.7220 - val_recall: 0.6169 - val_f1_score: 0.6653 -  
lr: 0.0010  
Epoch 14/100  
623/624 [=====>.] - ETA: 0s - loss: 0.6159 -  
accuracy: 0.7854 - precision: 0.8491 - recall: 0.7262 - f1_score:  
0.7829  
Epoch 14: val_accuracy improved from 0.70873 to 0.75220, saving model  
to models\modelo_S_com_data_augmentation_adam.keras  
624/624 [=====] - 33s 52ms/step - loss:  
0.6160 - accuracy: 0.7854 - precision: 0.8492 - recall: 0.7263 -  
f1_score: 0.7830 - val_loss: 0.7720 - val_accuracy: 0.7522 -  
val_precision: 0.8182 - val_recall: 0.7037 - val_f1_score: 0.7567 -  
lr: 5.0000e-04  
Epoch 15/100  
623/624 [=====>.] - ETA: 0s - loss: 0.5967 -  
accuracy: 0.7926 - precision: 0.8514 - recall: 0.7357 - f1_score:  
0.7893
```

Epoch 15: val_accuracy did not improve from 0.75220
624/624 [=====] - 32s 52ms/step - loss: 0.5974 - accuracy: 0.7924 - precision: 0.8513 - recall: 0.7356 - f1_score: 0.7892 - val_loss: 0.7612 - val_accuracy: 0.7490 - val_precision: 0.7991 - val_recall: 0.7057 - val_f1_score: 0.7495 - lr: 5.0000e-04
Epoch 16/100
623/624 [=====>.] - ETA: 0s - loss: 0.5869 - accuracy: 0.7931 - precision: 0.8525 - recall: 0.7394 - f1_score: 0.7919
Epoch 16: val_accuracy improved from 0.75220 to 0.78335, saving model to models\modelo_S_com_data_augmentation_adam.keras
624/624 [=====] - 33s 52ms/step - loss: 0.5868 - accuracy: 0.7932 - precision: 0.8526 - recall: 0.7395 - f1_score: 0.7920 - val_loss: 0.6476 - val_accuracy: 0.7834 - val_precision: 0.8406 - val_recall: 0.7339 - val_f1_score: 0.7836 - lr: 5.0000e-04
Epoch 17/100
624/624 [=====] - ETA: 0s - loss: 0.5644 - accuracy: 0.8002 - precision: 0.8562 - recall: 0.7487 - f1_score: 0.7989
Epoch 17: val_accuracy improved from 0.78335 to 0.78566, saving model to models\modelo_S_com_data_augmentation_adam.keras
624/624 [=====] - 33s 52ms/step - loss: 0.5644 - accuracy: 0.8002 - precision: 0.8562 - recall: 0.7487 - f1_score: 0.7989 - val_loss: 0.6252 - val_accuracy: 0.7857 - val_precision: 0.8405 - val_recall: 0.7421 - val_f1_score: 0.7882 - lr: 5.0000e-04
Epoch 18/100
624/624 [=====] - ETA: 0s - loss: 0.5551 - accuracy: 0.8050 - precision: 0.8607 - recall: 0.7552 - f1_score: 0.8045
Epoch 18: val_accuracy did not improve from 0.78566
624/624 [=====] - 34s 54ms/step - loss: 0.5551 - accuracy: 0.8050 - precision: 0.8607 - recall: 0.7552 - f1_score: 0.8045 - val_loss: 0.8144 - val_accuracy: 0.7459 - val_precision: 0.7968 - val_recall: 0.7088 - val_f1_score: 0.7502 - lr: 5.0000e-04
Epoch 19/100
623/624 [=====>.] - ETA: 0s - loss: 0.5477 - accuracy: 0.8074 - precision: 0.8620 - recall: 0.7607 - f1_score: 0.8082
Epoch 19: val_accuracy did not improve from 0.78566
624/624 [=====] - 33s 52ms/step - loss: 0.5477 - accuracy: 0.8075 - precision: 0.8620 - recall: 0.7607 - f1_score: 0.8082 - val_loss: 0.8062 - val_accuracy: 0.7353 - val_precision: 0.7881 - val_recall: 0.6938 - val_f1_score: 0.7379 - lr: 5.0000e-04
Epoch 20/100
624/624 [=====] - ETA: 0s - loss: 0.5324 -

accuracy: 0.8137 - precision: 0.8636 - recall: 0.7656 - f1_score: 0.8117

Epoch 20: val_accuracy did not improve from 0.78566

Epoch 20: ReduceLRonPlateau reducing learning rate to 0.0002500000118743628.

624/624 [=====] - 32s 52ms/step - loss: 0.5324 - accuracy: 0.8137 - precision: 0.8636 - recall: 0.7656 - f1_score: 0.8117 - val_loss: 0.6503 - val_accuracy: 0.7834 - val_precision: 0.8299 - val_recall: 0.7440 - val_f1_score: 0.7846 - lr: 5.0000e-04

Epoch 21/100

624/624 [=====] - ETA: 0s - loss: 0.5101 - accuracy: 0.8213 - precision: 0.8692 - recall: 0.7770 - f1_score: 0.8205

Epoch 21: val_accuracy improved from 0.78566 to 0.81941, saving model to models\modelo_S_com_data_augmentation_adam.keras

624/624 [=====] - 34s 54ms/step - loss: 0.5101 - accuracy: 0.8213 - precision: 0.8692 - recall: 0.7770 - f1_score: 0.8205 - val_loss: 0.5298 - val_accuracy: 0.8194 - val_precision: 0.8635 - val_recall: 0.7851 - val_f1_score: 0.8224 - lr: 2.5000e-04

Epoch 22/100

624/624 [=====] - ETA: 0s - loss: 0.4904 - accuracy: 0.8275 - precision: 0.8746 - recall: 0.7838 - f1_score: 0.8267

Epoch 22: val_accuracy did not improve from 0.81941

624/624 [=====] - 33s 53ms/step - loss: 0.4904 - accuracy: 0.8275 - precision: 0.8746 - recall: 0.7838 - f1_score: 0.8267 - val_loss: 0.5830 - val_accuracy: 0.8066 - val_precision: 0.8527 - val_recall: 0.7686 - val_f1_score: 0.8085 - lr: 2.5000e-04

Epoch 23/100

624/624 [=====] - ETA: 0s - loss: 0.4848 - accuracy: 0.8288 - precision: 0.8758 - recall: 0.7859 - f1_score: 0.8284

Epoch 23: val_accuracy did not improve from 0.81941

624/624 [=====] - 33s 53ms/step - loss: 0.4848 - accuracy: 0.8288 - precision: 0.8758 - recall: 0.7859 - f1_score: 0.8284 - val_loss: 0.6124 - val_accuracy: 0.7964 - val_precision: 0.8386 - val_recall: 0.7634 - val_f1_score: 0.7992 - lr: 2.5000e-04

Epoch 24/100

623/624 [=====>.] - ETA: 0s - loss: 0.4737 - accuracy: 0.8348 - precision: 0.8783 - recall: 0.7940 - f1_score: 0.8340

Epoch 24: val_accuracy did not improve from 0.81941

Epoch 24: ReduceLRonPlateau reducing learning rate to 0.0001250000059371814.

```
624/624 [=====] - 33s 53ms/step - loss:
0.4737 - accuracy: 0.8348 - precision: 0.8783 - recall: 0.7940 -
f1_score: 0.8340 - val_loss: 0.5317 - val_accuracy: 0.8186 -
val_precision: 0.8635 - val_recall: 0.7862 - val_f1_score: 0.8230 -
lr: 2.5000e-04
Epoch 25/100
623/624 [=====>.] - ETA: 0s - loss: 0.4590 -
accuracy: 0.8390 - precision: 0.8811 - recall: 0.7992 - f1_score:
0.8381
Epoch 25: val_accuracy did not improve from 0.81941
624/624 [=====] - 33s 53ms/step - loss:
0.4588 - accuracy: 0.8391 - precision: 0.8812 - recall: 0.7993 -
f1_score: 0.8382 - val_loss: 0.5647 - val_accuracy: 0.8150 -
val_precision: 0.8550 - val_recall: 0.7817 - val_f1_score: 0.8167 -
lr: 1.2500e-04
Epoch 26/100
624/624 [=====] - ETA: 0s - loss: 0.4485 -
accuracy: 0.8404 - precision: 0.8841 - recall: 0.8028 - f1_score:
0.8415
Epoch 26: val_accuracy improved from 0.81941 to 0.83894, saving model
to models\modelo_S_com_data_augmentation_adam.keras
624/624 [=====] - 33s 53ms/step - loss:
0.4485 - accuracy: 0.8404 - precision: 0.8841 - recall: 0.8028 -
f1_score: 0.8415 - val_loss: 0.4765 - val_accuracy: 0.8389 -
val_precision: 0.8798 - val_recall: 0.8079 - val_f1_score: 0.8423 -
lr: 1.2500e-04
Epoch 27/100
623/624 [=====>.] - ETA: 0s - loss: 0.4498 -
accuracy: 0.8425 - precision: 0.8850 - recall: 0.8033 - f1_score:
0.8422
Epoch 27: val_accuracy did not improve from 0.83894
624/624 [=====] - 33s 53ms/step - loss:
0.4502 - accuracy: 0.8425 - precision: 0.8849 - recall: 0.8032 -
f1_score: 0.8421 - val_loss: 0.5164 - val_accuracy: 0.8256 -
val_precision: 0.8675 - val_recall: 0.7941 - val_f1_score: 0.8292 -
lr: 1.2500e-04
Epoch 28/100
624/624 [=====] - ETA: 0s - loss: 0.4444 -
accuracy: 0.8439 - precision: 0.8860 - recall: 0.8064 - f1_score:
0.8444
Epoch 28: val_accuracy improved from 0.83894 to 0.84095, saving model
to models\modelo_S_com_data_augmentation_adam.keras
624/624 [=====] - 33s 53ms/step - loss:
0.4444 - accuracy: 0.8439 - precision: 0.8860 - recall: 0.8064 -
f1_score: 0.8444 - val_loss: 0.4683 - val_accuracy: 0.8409 -
val_precision: 0.8782 - val_recall: 0.8080 - val_f1_score: 0.8416 -
lr: 1.2500e-04
Epoch 29/100
623/624 [=====>.] - ETA: 0s - loss: 0.4370 -
accuracy: 0.8468 - precision: 0.8871 - recall: 0.8101 - f1_score:
```

```
0.8468
Epoch 29: val_accuracy did not improve from 0.84095
624/624 [=====] - 33s 52ms/step - loss:
0.4370 - accuracy: 0.8468 - precision: 0.8870 - recall: 0.8101 -
f1_score: 0.8468 - val_loss: 0.5098 - val_accuracy: 0.8290 -
val_precision: 0.8667 - val_recall: 0.8007 - val_f1_score: 0.8324 -
lr: 1.2500e-04
Epoch 30/100
624/624 [=====] - ETA: 0s - loss: 0.4317 -
accuracy: 0.8485 - precision: 0.8885 - recall: 0.8136 - f1_score:
0.8494
Epoch 30: val_accuracy did not improve from 0.84095
624/624 [=====] - 33s 52ms/step - loss:
0.4317 - accuracy: 0.8485 - precision: 0.8885 - recall: 0.8136 -
f1_score: 0.8494 - val_loss: 0.4960 - val_accuracy: 0.8333 -
val_precision: 0.8694 - val_recall: 0.8052 - val_f1_score: 0.8360 -
lr: 1.2500e-04
Epoch 31/100
624/624 [=====] - ETA: 0s - loss: 0.4264 -
accuracy: 0.8513 - precision: 0.8891 - recall: 0.8157 - f1_score:
0.8508
Epoch 31: val_accuracy did not improve from 0.84095

Epoch 31: ReduceLROnPlateau reducing learning rate to
6.25000029685907e-05.
624/624 [=====] - 32s 52ms/step - loss:
0.4264 - accuracy: 0.8513 - precision: 0.8891 - recall: 0.8157 -
f1_score: 0.8508 - val_loss: 0.5034 - val_accuracy: 0.8332 -
val_precision: 0.8702 - val_recall: 0.8038 - val_f1_score: 0.8357 -
lr: 1.2500e-04
Epoch 32/100
623/624 [=====>.] - ETA: 0s - loss: 0.4224 -
accuracy: 0.8514 - precision: 0.8903 - recall: 0.8158 - f1_score:
0.8514
Epoch 32: val_accuracy did not improve from 0.84095
624/624 [=====] - 33s 52ms/step - loss:
0.4222 - accuracy: 0.8515 - precision: 0.8904 - recall: 0.8158 -
f1_score: 0.8515 - val_loss: 0.4868 - val_accuracy: 0.8371 -
val_precision: 0.8739 - val_recall: 0.8071 - val_f1_score: 0.8392 -
lr: 6.2500e-05
Epoch 33/100
624/624 [=====] - ETA: 0s - loss: 0.4143 -
accuracy: 0.8561 - precision: 0.8947 - recall: 0.8217 - f1_score:
0.8566
Epoch 33: val_accuracy did not improve from 0.84095
624/624 [=====] - 32s 52ms/step - loss:
0.4143 - accuracy: 0.8561 - precision: 0.8947 - recall: 0.8217 -
f1_score: 0.8566 - val_loss: 0.4835 - val_accuracy: 0.8397 -
val_precision: 0.8741 - val_recall: 0.8100 - val_f1_score: 0.8408 -
lr: 6.2500e-05
```

Epoch 34/100
623/624 [=====>.] - ETA: 0s - loss: 0.4124 - accuracy: 0.8563 - precision: 0.8935 - recall: 0.8212 - f1_score: 0.8558
Epoch 34: val_accuracy did not improve from 0.84095

Epoch 34: ReduceLRonPlateau reducing learning rate to 3.125000148429535e-05.
624/624 [=====] - 33s 52ms/step - loss: 0.4125 - accuracy: 0.8561 - precision: 0.8934 - recall: 0.8210 - f1_score: 0.8557 - val_loss: 0.4861 - val_accuracy: 0.8378 - val_precision: 0.8720 - val_recall: 0.8075 - val_f1_score: 0.8385 - lr: 6.2500e-05
Epoch 35/100
624/624 [=====] - ETA: 0s - loss: 0.4080 - accuracy: 0.8580 - precision: 0.8927 - recall: 0.8232 - f1_score: 0.8566
Epoch 35: val_accuracy improved from 0.84095 to 0.84565, saving model to models\modelo_S_com_data_augmentation_adam.keras
624/624 [=====] - 33s 53ms/step - loss: 0.4080 - accuracy: 0.8580 - precision: 0.8927 - recall: 0.8232 - f1_score: 0.8566 - val_loss: 0.4550 - val_accuracy: 0.8457 - val_precision: 0.8807 - val_recall: 0.8161 - val_f1_score: 0.8472 - lr: 3.1250e-05
Epoch 36/100
623/624 [=====>.] - ETA: 0s - loss: 0.4093 - accuracy: 0.8566 - precision: 0.8932 - recall: 0.8243 - f1_score: 0.8574
Epoch 36: val_accuracy did not improve from 0.84565
624/624 [=====] - 33s 53ms/step - loss: 0.4093 - accuracy: 0.8566 - precision: 0.8932 - recall: 0.8243 - f1_score: 0.8574 - val_loss: 0.4811 - val_accuracy: 0.8397 - val_precision: 0.8737 - val_recall: 0.8095 - val_f1_score: 0.8404 - lr: 3.1250e-05
Epoch 37/100
623/624 [=====>.] - ETA: 0s - loss: 0.4076 - accuracy: 0.8562 - precision: 0.8931 - recall: 0.8220 - f1_score: 0.8561
Epoch 37: val_accuracy did not improve from 0.84565
624/624 [=====] - 33s 53ms/step - loss: 0.4075 - accuracy: 0.8562 - precision: 0.8931 - recall: 0.8220 - f1_score: 0.8561 - val_loss: 0.4680 - val_accuracy: 0.8415 - val_precision: 0.8764 - val_recall: 0.8128 - val_f1_score: 0.8434 - lr: 3.1250e-05
Epoch 38/100
623/624 [=====>.] - ETA: 0s - loss: 0.4074 - accuracy: 0.8569 - precision: 0.8941 - recall: 0.8252 - f1_score: 0.8582
Epoch 38: val_accuracy did not improve from 0.84565

Epoch 38: ReduceLROnPlateau reducing learning rate to 1.5625000742147677e-05.
624/624 [=====] - 32s 52ms/step - loss: 0.4074 - accuracy: 0.8568 - precision: 0.8941 - recall: 0.8251 - f1_score: 0.8582 - val_loss: 0.4960 - val_accuracy: 0.8347 - val_precision: 0.8707 - val_recall: 0.8051 - val_f1_score: 0.8366 - lr: 3.1250e-05
Epoch 39/100
624/624 [=====] - ETA: 0s - loss: 0.4023 - accuracy: 0.8563 - precision: 0.8940 - recall: 0.8242 - f1_score: 0.8577
Epoch 39: val_accuracy did not improve from 0.84565
624/624 [=====] - 32s 52ms/step - loss: 0.4023 - accuracy: 0.8563 - precision: 0.8940 - recall: 0.8242 - f1_score: 0.8577 - val_loss: 0.4667 - val_accuracy: 0.8419 - val_precision: 0.8771 - val_recall: 0.8125 - val_f1_score: 0.8436 - lr: 1.5625e-05
Epoch 40/100
623/624 [=====>.] - ETA: 0s - loss: 0.4054 - accuracy: 0.8591 - precision: 0.8949 - recall: 0.8250 - f1_score: 0.8585
Epoch 40: val_accuracy did not improve from 0.84565
624/624 [=====] - 33s 52ms/step - loss: 0.4052 - accuracy: 0.8592 - precision: 0.8949 - recall: 0.8250 - f1_score: 0.8586 - val_loss: 0.4813 - val_accuracy: 0.8407 - val_precision: 0.8721 - val_recall: 0.8095 - val_f1_score: 0.8396 - lr: 1.5625e-05
Epoch 41/100
624/624 [=====] - ETA: 0s - loss: 0.3980 - accuracy: 0.8601 - precision: 0.8952 - recall: 0.8250 - f1_score: 0.8586
Epoch 41: val_accuracy did not improve from 0.84565
Epoch 41: ReduceLROnPlateau reducing learning rate to 7.812500371073838e-06.
624/624 [=====] - 32s 52ms/step - loss: 0.3980 - accuracy: 0.8601 - precision: 0.8952 - recall: 0.8250 - f1_score: 0.8586 - val_loss: 0.4651 - val_accuracy: 0.8431 - val_precision: 0.8772 - val_recall: 0.8130 - val_f1_score: 0.8439 - lr: 1.5625e-05
Epoch 42/100
623/624 [=====>.] - ETA: 0s - loss: 0.4068 - accuracy: 0.8565 - precision: 0.8932 - recall: 0.8237 - f1_score: 0.8570
Epoch 42: val_accuracy did not improve from 0.84565
624/624 [=====] - 32s 52ms/step - loss: 0.4066 - accuracy: 0.8565 - precision: 0.8932 - recall: 0.8238 - f1_score: 0.8571 - val_loss: 0.4732 - val_accuracy: 0.8419 - val_precision: 0.8754 - val_recall: 0.8125 - val_f1_score: 0.8428 -


```

lr: 7.8125e-06
Epoch 43/100
624/624 [=====] - ETA: 0s - loss: 0.4079 -
accuracy: 0.8567 - precision: 0.8915 - recall: 0.8253 - f1_score:
0.8571
Epoch 43: val_accuracy did not improve from 0.84565
624/624 [=====] - 33s 53ms/step - loss:
0.4079 - accuracy: 0.8567 - precision: 0.8915 - recall: 0.8253 -
f1_score: 0.8571 - val_loss: 0.4705 - val_accuracy: 0.8424 -
val_precision: 0.8752 - val_recall: 0.8127 - val_f1_score: 0.8428 -
lr: 7.8125e-06
Epoch 44/100
623/624 [=====>.] - ETA: 0s - loss: 0.4043 -
accuracy: 0.8574 - precision: 0.8949 - recall: 0.8246 - f1_score:
0.8583
Epoch 44: val_accuracy did not improve from 0.84565

Epoch 44: ReduceLROnPlateau reducing learning rate to
3.906250185536919e-06.
624/624 [=====] - 32s 52ms/step - loss:
0.4046 - accuracy: 0.8573 - precision: 0.8948 - recall: 0.8246 -
f1_score: 0.8583 - val_loss: 0.4751 - val_accuracy: 0.8412 -
val_precision: 0.8749 - val_recall: 0.8115 - val_f1_score: 0.8420 -
lr: 7.8125e-06
Epoch 45/100
623/624 [=====>.] - ETA: 0s - loss: 0.3968 -
accuracy: 0.8594 - precision: 0.8966 - recall: 0.8275 - f1_score:
0.8606
Epoch 45: val_accuracy did not improve from 0.84565
624/624 [=====] - 33s 52ms/step - loss:
0.3970 - accuracy: 0.8594 - precision: 0.8966 - recall: 0.8275 -
f1_score: 0.8607 - val_loss: 0.4668 - val_accuracy: 0.8434 -
val_precision: 0.8766 - val_recall: 0.8140 - val_f1_score: 0.8441 -
lr: 3.9063e-06
157/157 [=====] - 6s 40ms/step - loss: 0.4496
- accuracy: 0.8505 - precision: 0.8816 - recall: 0.8231 - f1_score:
0.8514
Test Loss: 0.4495638310909271
Test Accuracy: 0.8504999876022339
Test Precision: 0.8816409707069397
Test Recall: 0.8230999708175659
Test F1 Score: 0.851365327835083

```

Podemos ver acima que existe melhora mesmo que pouco significativa quando comparado à versão sem data augmentation.

```

# Plots do treino
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_S_com_data_augmentation_adam.png')
plt.tight_layout()

```

