04_model_no_data_augmentation.pdf

January 11, 2026

[1]:
```
import tensorflow as tf
from tensorflow import keras
from keras import layers, models
import matplotlib.pyplot as plt
import numpy as np

import sys
from pathlib import Path
from keras import mixed_precision
# Enables the card's high-speed FP16 cores
policy = mixed_precision.Policy('mixed_float16')
mixed_precision.set_global_policy(policy)

PROJECT_ROOT = Path.cwd().parent
if str(PROJECT_ROOT) not in sys.path:
    sys.path.insert(0, str(PROJECT_ROOT))


from src import dataset

print("Physical Devices:", tf.config.list_physical_devices())
print("Num GPUs Available: ", len(tf.config.list_physical_devices('GPU')))

tf.random.set_seed(42)
np.random.seed(42)
```

```
2026-01-11 20:35:18.142050: 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: SSE3 SSE4.1 SSE4.2 AVX AVX2 AVX512F
AVX512_VNNI AVX512_BF16 FMA, in other operations, rebuild TensorFlow with the
appropriate compiler flags.
```

```
Physical Devices: [PhysicalDevice(name='/physical_device:CPU:0',
device_type='CPU'), PhysicalDevice(name='/physical_device:GPU:0',
device_type='GPU')]
Num GPUs Available:  1
```

```
2026-01-11 20:35:19.642959: I
```

```
external/local_xla/xla/stream_executor/rocm/rocm_executor.cc:920] successful
NUMA node read from SysFS had negative value (-1), but there must be at least
one NUMA node, so returning NUMA node zero
2026-01-11 20:35:19.643010: I
external/local_xla/xla/stream_executor/rocm/rocm_executor.cc:920] successful
NUMA node read from SysFS had negative value (-1), but there must be at least
one NUMA node, so returning NUMA node zero
2026-01-11 20:35:20.052676: I
external/local_xla/xla/stream_executor/rocm/rocm_executor.cc:920] successful
NUMA node read from SysFS had negative value (-1), but there must be at least
one NUMA node, so returning NUMA node zero
2026-01-11 20:35:20.052720: I
external/local_xla/xla/stream_executor/rocm/rocm_executor.cc:920] successful
NUMA node read from SysFS had negative value (-1), but there must be at least
one NUMA node, so returning NUMA node zero
2026-01-11 20:35:20.052744: I
external/local_xla/xla/stream_executor/rocm/rocm_executor.cc:920] successful
NUMA node read from SysFS had negative value (-1), but there must be at least
one NUMA node, so returning NUMA node zero
2026-01-11 20:35:20.052764: I
external/local_xla/xla/stream_executor/rocm/rocm_executor.cc:920] successful
NUMA node read from SysFS had negative value (-1), but there must be at least
one NUMA node, so returning NUMA node zero
2026-01-11 20:35:20.052773: I
tensorflow/core/common_runtime/gpu/gpu_device.cc:2397] Ignoring visible gpu
device (device: 1, name: AMD Radeon Graphics, pci bus id: 0000:0e:00.0) with
AMDGPU version : gfx1036. The supported AMDGPU versions are gfx900, gfx906,
gfx908, gfx90a, gfx942, gfx1030, gfx1100, gfx1101, gfx1102, gfx1200, gfx1201.
```

[2]:
```
IMAGE_HEIGHT = 256
IMAGE_WIDTH = 256
BATCH_SIZE = 256
```

[3]:
```
train_ds, val_ds, test_ds = dataset.
  ↪create_datasets(IMAGE_HEIGHT,IMAGE_WIDTH,BATCH_SIZE)
```

```
Number of images:  16970
Attributes:  ['id', 'width', 'height', 'file_name', 'license', 'flickr_url',
'coco_url', 'date_captured']

Number of annotations (they refer to images): 16970
Attributes:  ['id', 'image_id', 'category_id', 'segmentation', 'area', 'bbox',
'iscrowd', 'attributes']

IDs don't always match!
17011 != 17000
Size of paths and labels should be the same:  16970 16970 0
Training samples: 10182
```

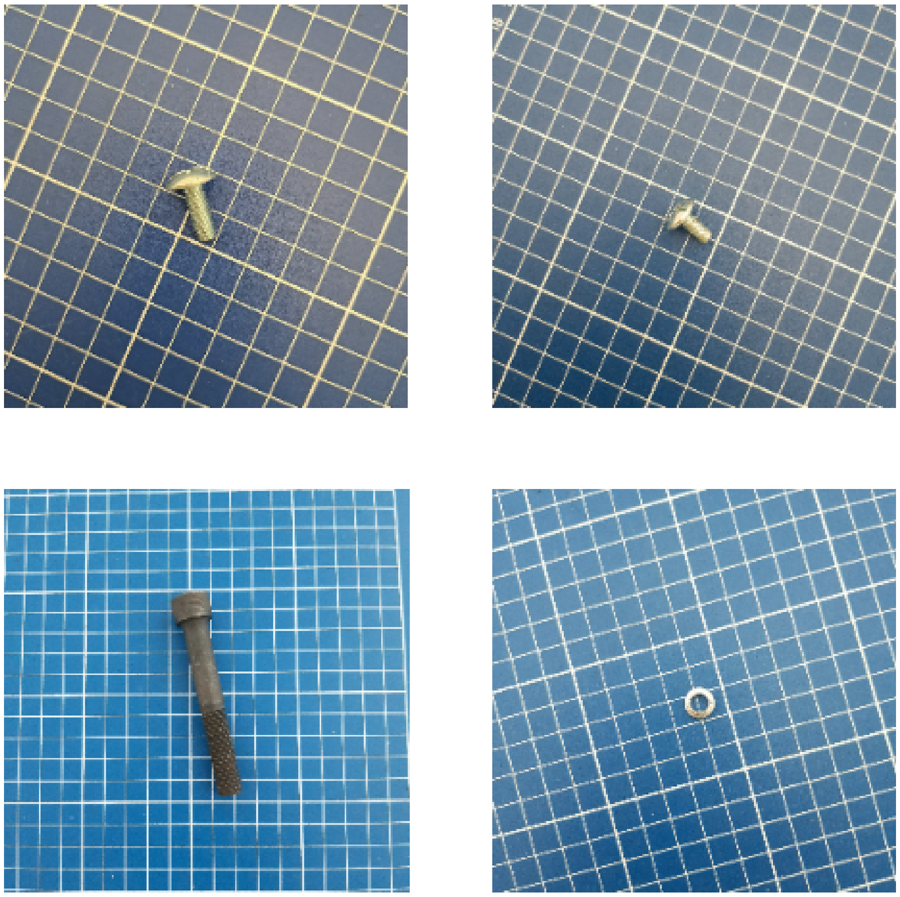
```
Validation samples: 3394
Testing samples: 3394
```

```
2026-01-11 20:35:31.672750: I
external/local_xla/xla/stream_executor/rocm/rocm_executor.cc:920] successful
NUMA node read from SysFS had negative value (-1), but there must be at least
one NUMA node, so returning NUMA node zero
2026-01-11 20:35:31.672866: I
external/local_xla/xla/stream_executor/rocm/rocm_executor.cc:920] successful
NUMA node read from SysFS had negative value (-1), but there must be at least
one NUMA node, so returning NUMA node zero
2026-01-11 20:35:31.672896: I
external/local_xla/xla/stream_executor/rocm/rocm_executor.cc:920] successful
NUMA node read from SysFS had negative value (-1), but there must be at least
one NUMA node, so returning NUMA node zero
2026-01-11 20:35:31.672995: I
external/local_xla/xla/stream_executor/rocm/rocm_executor.cc:920] successful
NUMA node read from SysFS had negative value (-1), but there must be at least
one NUMA node, so returning NUMA node zero
2026-01-11 20:35:31.673044: I
external/local_xla/xla/stream_executor/rocm/rocm_executor.cc:920] successful
NUMA node read from SysFS had negative value (-1), but there must be at least
one NUMA node, so returning NUMA node zero
2026-01-11 20:35:31.673090: I
external/local_xla/xla/stream_executor/rocm/rocm_executor.cc:920] successful
NUMA node read from SysFS had negative value (-1), but there must be at least
one NUMA node, so returning NUMA node zero
2026-01-11 20:35:31.673111: I
tensorflow/core/common_runtime/gpu/gpu_device.cc:2021] Created device
/job:localhost/replica:0/task:0/device:GPU:0 with 14748 MB memory:  -> device:
0, name: AMD Radeon RX 9070 XT, pci bus id: 0000:03:00.0
```

[4]:
```
for images, labels in train_ds.take(1):
    plt.figure(figsize=(6,6))
    for i in range(4):
        plt.subplot(2,2,i+1)
        plt.imshow(images[i].numpy())
        plt.axis("off")
    plt.show()
```

```
2026-01-11 20:35:42.461169: I
tensorflow/core/kernels/data/shuffle_dataset_op.cc:450] ShuffleDatasetV3:3:
Filling up shuffle buffer (this may take a while): 1744 of 10182
2026-01-11 20:35:52.468445: I
tensorflow/core/kernels/data/shuffle_dataset_op.cc:450] ShuffleDatasetV3:3:
Filling up shuffle buffer (this may take a while): 3350 of 10182
2026-01-11 20:36:12.455107: I
tensorflow/core/kernels/data/shuffle_dataset_op.cc:450] ShuffleDatasetV3:3:
Filling up shuffle buffer (this may take a while): 6588 of 10182
```

```
2026-01-11 20:36:22.459252: I
tensorflow/core/kernels/data/shuffle_dataset_op.cc:450] ShuffleDatasetV3:3:
Filling up shuffle buffer (this may take a while): 8184 of 10182
2026-01-11 20:36:32.489323: I
tensorflow/core/kernels/data/shuffle_dataset_op.cc:450] ShuffleDatasetV3:3:
Filling up shuffle buffer (this may take a while): 9813 of 10182
2026-01-11 20:36:34.710649: I
tensorflow/core/kernels/data/shuffle_dataset_op.cc:480] Shuffle buffer filled.
```



```
2026-01-11 20:36:34.833684: I tensorflow/core/framework/local_rendezvous.cc:404]
Local rendezvous is aborting with status: OUT_OF_RANGE: End of sequence
```

[5]:
```
data_augmentation = tf.keras.Sequential([
        layers.RandomFlip("horizontal"),
        layers.RandomFlip("vertical"),
        layers.RandomRotation(0.2),
        layers.RandomZoom(0.2),
    ])
```

```python
def residual_block(x, filters, stride=1):
    shortcut = x

    x = layers.Conv2D(filters, 3, strides=stride, padding="same",
                      kernel_regularizer=keras.regularizers.l2(1e-4))(x)
    x = layers.BatchNormalization()(x)
    x = layers.Activation("relu")(x)

    x = layers.Conv2D(filters, 3, padding="same",
                      kernel_regularizer=keras.regularizers.l2(1e-4))(x)
    x = layers.BatchNormalization()(x)

    if stride != 1 or shortcut.shape[-1] != filters:
        shortcut = layers.Conv2D(filters, 1, strides=stride,
                                 kernel_regularizer=keras.regularizers.
 ↪l2(1e-4))(shortcut)
        shortcut = layers.BatchNormalization()(shortcut)

    x = layers.Add()([x, shortcut])
    x = layers.Activation("relu")(x)
    return x


def build_good_model(num_classes,IMAGE_WIDTH,IMAGE_HEIGHT):
    inputs = keras.Input(shape=(IMAGE_WIDTH, IMAGE_HEIGHT, 3))

    # x = data_augmentation(inputs)
    # MOJA GRAFICKA NE PODUPIRE OVAKVE NAREDBE PA SE IZVODI NA PROCESORU!!!!!!!!
 ↪!!!!!!

    x = layers.Rescaling(1./255)(inputs)

    x = layers.Conv2D(32, 3, padding="same")(x)
    x = layers.BatchNormalization()(x)
    x = layers.Activation("relu")(x)

    x = residual_block(x, 64, stride=2)
    x = residual_block(x, 64)

    x = residual_block(x, 128, stride=2)
    x = residual_block(x, 128)

    x = residual_block(x, 256, stride=2)
    x = residual_block(x, 256)

    x = residual_block(x, 512, stride=2)
    x = layers.GlobalAveragePooling2D()(x)
    x = layers.Dropout(0.4)(x)
    outputs = layers.Dense(num_classes, activation="softmax",dtype='float32')(x)

    model = keras.Model(inputs=inputs, outputs=outputs, name="good_model")
    return model
```

[6]:
```
model = build_good_model(6,IMAGE_WIDTH,IMAGE_HEIGHT)
model.summary()
```

```
Model: "good_model"


 Layer (type)          Output Shape          Param #  Connected to

 input_layer           (None, 256, 256,            0  -
 (InputLayer)          3)

 rescaling             (None, 256, 256,            0  input_layer[0][0]
 (Rescaling)           3)

 conv2d (Conv2D)       (None, 256, 256,          896  rescaling[0][0]
                       32)

 batch_normalization   (None, 256, 256,          128  conv2d[0][0]
 (BatchNormalizatio…   32)

 activation            (None, 256, 256,            0  batch_normalizat…
 (Activation)          32)

 conv2d_1 (Conv2D)     (None, 128, 128,       18,496  activation[0][0]
                       64)

 batch_normalizatio…   (None, 128, 128,          256  conv2d_1[0][0]
 (BatchNormalizatio…   64)

 activation_1          (None, 128, 128,            0  batch_normalizat…
 (Activation)          64)

 conv2d_2 (Conv2D)     (None, 128, 128,       36,928  activation_1[0][…
                       64)

 conv2d_3 (Conv2D)     (None, 128, 128,        2,112  activation[0][0]
                       64)

 batch_normalizatio…   (None, 128, 128,          256  conv2d_2[0][0]
 (BatchNormalizatio…   64)
```

```
batch_normalizatio…   (None, 128, 128,          256   conv2d_3[0][0]
(BatchNormalizatio…   64)

add (Add)             (None, 128, 128,            0   batch_normalizat…
                      64)                             batch_normalizat…

activation_2          (None, 128, 128,            0   add[0][0]
(Activation)          64)

conv2d_4 (Conv2D)     (None, 128, 128,       36,928   activation_2[0][…
                      64)

batch_normalizatio…   (None, 128, 128,          256   conv2d_4[0][0]
(BatchNormalizatio…   64)

activation_3          (None, 128, 128,            0   batch_normalizat…
(Activation)          64)

conv2d_5 (Conv2D)     (None, 128, 128,       36,928   activation_3[0][…
                      64)

batch_normalizatio…   (None, 128, 128,          256   conv2d_5[0][0]
(BatchNormalizatio…   64)

add_1 (Add)           (None, 128, 128,            0   batch_normalizat…
                      64)                             activation_2[0][…

activation_4          (None, 128, 128,            0   add_1[0][0]
(Activation)          64)

conv2d_6 (Conv2D)     (None, 64, 64,         73,856   activation_4[0][…
                      128)

batch_normalizatio…   (None, 64, 64,            512   conv2d_6[0][0]
(BatchNormalizatio…   128)

activation_5          (None, 64, 64,              0   batch_normalizat…
(Activation)          128)

conv2d_7 (Conv2D)     (None, 64, 64,        147,584   activation_5[0][…
                      128)

conv2d_8 (Conv2D)     (None, 64, 64,          8,320   activation_4[0][…
                      128)

batch_normalizatio…   (None, 64, 64,            512   conv2d_7[0][0]
(BatchNormalizatio…   128)
```

```
batch_normalizatio…   (None, 64, 64,              512   conv2d_8[0][0]
(BatchNormalizatio…   128)

add_2 (Add)           (None, 64, 64,                0   batch_normalizat…
                      128)                              batch_normalizat…

activation_6          (None, 64, 64,                0   add_2[0][0]
(Activation)          128)

conv2d_9 (Conv2D)     (None, 64, 64,          147,584   activation_6[0][…
                      128)

batch_normalizatio…   (None, 64, 64,              512   conv2d_9[0][0]
(BatchNormalizatio…   128)

activation_7          (None, 64, 64,                0   batch_normalizat…
(Activation)          128)

conv2d_10 (Conv2D)    (None, 64, 64,          147,584   activation_7[0][…
                      128)

batch_normalizatio…   (None, 64, 64,              512   conv2d_10[0][0]
(BatchNormalizatio…   128)

add_3 (Add)           (None, 64, 64,                0   batch_normalizat…
                      128)                              activation_6[0][…

activation_8          (None, 64, 64,                0   add_3[0][0]
(Activation)          128)

conv2d_11 (Conv2D)    (None, 32, 32,          295,168   activation_8[0][…
                      256)

batch_normalizatio…   (None, 32, 32,            1,024   conv2d_11[0][0]
(BatchNormalizatio…   256)

activation_9          (None, 32, 32,                0   batch_normalizat…
(Activation)          256)

conv2d_12 (Conv2D)    (None, 32, 32,          590,080   activation_9[0][…
                      256)

conv2d_13 (Conv2D)    (None, 32, 32,           33,024   activation_8[0][…
                      256)

batch_normalizatio…   (None, 32, 32,            1,024   conv2d_12[0][0]
(BatchNormalizatio…   256)
```

```
batch_normalizatio…   (None, 32, 32,          1,024   conv2d_13[0][0]
(BatchNormalizatio…   256)

add_4 (Add)           (None, 32, 32,              0   batch_normalizat…
                      256)                            batch_normalizat…

activation_10         (None, 32, 32,              0   add_4[0][0]
(Activation)          256)

conv2d_14 (Conv2D)    (None, 32, 32,        590,080   activation_10[0]…
                      256)

batch_normalizatio…   (None, 32, 32,          1,024   conv2d_14[0][0]
(BatchNormalizatio…   256)

activation_11         (None, 32, 32,              0   batch_normalizat…
(Activation)          256)

conv2d_15 (Conv2D)    (None, 32, 32,        590,080   activation_11[0]…
                      256)

batch_normalizatio…   (None, 32, 32,          1,024   conv2d_15[0][0]
(BatchNormalizatio…   256)

add_5 (Add)           (None, 32, 32,              0   batch_normalizat…
                      256)                            activation_10[0]…

activation_12         (None, 32, 32,              0   add_5[0][0]
(Activation)          256)

conv2d_16 (Conv2D)    (None, 16, 16,      1,180,160   activation_12[0]…
                      512)

batch_normalizatio…   (None, 16, 16,          2,048   conv2d_16[0][0]
(BatchNormalizatio…   512)

activation_13         (None, 16, 16,              0   batch_normalizat…
(Activation)          512)

conv2d_17 (Conv2D)    (None, 16, 16,      2,359,808   activation_13[0]…
                      512)

conv2d_18 (Conv2D)    (None, 16, 16,        131,584   activation_12[0]…
                      512)

batch_normalizatio…   (None, 16, 16,          2,048   conv2d_17[0][0]
(BatchNormalizatio…   512)
```

```
batch_normalizatio…   (None, 16, 16,           2,048   conv2d_18[0][0]
(BatchNormalizatio…   512)

add_6 (Add)           (None, 16, 16,               0   batch_normalizat…
                      512)                             batch_normalizat…

activation_14         (None, 16, 16,               0   add_6[0][0]
(Activation)          512)

global_average_poo…   (None, 512)                  0   activation_14[0]…
(GlobalAveragePool…

dropout (Dropout)     (None, 512)                  0   global_average_p…

dense (Dense)         (None, 6)                3,078   dropout[0][0]


Total params: 6,445,510 (24.59 MB)

Trainable params: 6,437,894 (24.56 MB)

Non-trainable params: 7,616 (29.75 KB)
```

[7]:
```
model.compile(
    optimizer=tf.keras.optimizers.Adam(learning_rate=1e-4), # Smanjeno s 0.001␣
↪na 0.0001 kako ne bi loss eksplodirao i postao NaN
    loss="sparse_categorical_crossentropy",
    metrics=["accuracy"],
    jit_compile=True,          # <--- FIX 1: Fuses GPU operations together
    steps_per_execution=50,    # <--- FIX 2: Sends 50 batches to GPU at once
)

lr_cb = keras.callbacks.ReduceLROnPlateau(
    monitor="val_loss",
    factor=0.3,
    patience=5,
    min_lr=1e-6
)

early_cb = keras.callbacks.EarlyStopping(
    monitor="val_loss",
    patience=10,
    restore_best_weights=True
)
```

[8]:
```
history = model.fit(
    train_ds,
    validation_data=val_ds,
    epochs=50,
    callbacks = [lr_cb,early_cb]
)
```

```
Epoch 1/50

WARNING: All log messages before absl::InitializeLog() is called are written to
STDERR
I0000 00:00:1768160200.726503   53514 service.cc:146] XLA service 0x79d164012910
initialized for platform ROCM (this does not guarantee that XLA will be used).
Devices:
I0000 00:00:1768160200.726527   53514 service.cc:154]   StreamExecutor device
(0): AMD Radeon RX 9070 XT, AMDGPU ISA version: gfx1201
2026-01-11 20:36:40.834549: I
tensorflow/compiler/mlir/tensorflow/utils/dump_mlir_util.cc:268] disabling MLIR
crash reproducer, set env var `MLIR_CRASH_REPRODUCER_DIRECTORY` to enable.
I0000 00:00:1768160206.712688   53514 device_compiler.h:188] Compiled cluster
using XLA!  This line is logged at most once for the lifetime of the process.

40/40            78s 2s/step -
accuracy: 0.4394 - loss: 1.8281 - val_accuracy: 0.2404 - val_loss: 2.4299 -
learning_rate: 1.0000e-04
Epoch 2/50
40/40            41s 1s/step -
accuracy: 0.5929 - loss: 1.4647 - val_accuracy: 0.1061 - val_loss: 2.7703 -
learning_rate: 1.0000e-04
Epoch 3/50
40/40            41s 1s/step -
accuracy: 0.6991 - loss: 1.1962 - val_accuracy: 0.1061 - val_loss: 2.6331 -
learning_rate: 1.0000e-04
Epoch 4/50
40/40            41s 1s/step -
accuracy: 0.7704 - loss: 0.9921 - val_accuracy: 0.1061 - val_loss: 4.3695 -
learning_rate: 1.0000e-04
Epoch 5/50
40/40            41s 1s/step -
accuracy: 0.8198 - loss: 0.8463 - val_accuracy: 0.1061 - val_loss: 2.5366 -
learning_rate: 1.0000e-04
Epoch 6/50
40/40            41s 1s/step -
accuracy: 0.8552 - loss: 0.7438 - val_accuracy: 0.1011 - val_loss: 2.9576 -
learning_rate: 1.0000e-04
Epoch 7/50
40/40            41s 1s/step -
accuracy: 0.8866 - loss: 0.6493 - val_accuracy: 0.1022 - val_loss: 2.4519 -
learning_rate: 3.0000e-05
```

```
Epoch 8/50
40/40            41s 1s/step -
accuracy: 0.9046 - loss: 0.6049 - val_accuracy: 0.1025 - val_loss: 2.4983 -
learning_rate: 3.0000e-05
Epoch 9/50
40/40            41s 1s/step -
accuracy: 0.9167 - loss: 0.5735 - val_accuracy: 0.1105 - val_loss: 2.4024 -
learning_rate: 3.0000e-05
Epoch 10/50
40/40            41s 1s/step -
accuracy: 0.9272 - loss: 0.5482 - val_accuracy: 0.1426 - val_loss: 2.4572 -
learning_rate: 3.0000e-05
Epoch 11/50
40/40            41s 1s/step -
accuracy: 0.9355 - loss: 0.5231 - val_accuracy: 0.2060 - val_loss: 2.2985 -
learning_rate: 3.0000e-05
Epoch 12/50
40/40            41s 1s/step -
accuracy: 0.9464 - loss: 0.4956 - val_accuracy: 0.2619 - val_loss: 2.2214 -
learning_rate: 3.0000e-05
Epoch 13/50
40/40            41s 1s/step -
accuracy: 0.9557 - loss: 0.4701 - val_accuracy: 0.4119 - val_loss: 2.0035 -
learning_rate: 3.0000e-05
Epoch 14/50
40/40            41s 1s/step -
accuracy: 0.9636 - loss: 0.4517 - val_accuracy: 0.4673 - val_loss: 1.8192 -
learning_rate: 3.0000e-05
Epoch 15/50
40/40            41s 1s/step -
accuracy: 0.9697 - loss: 0.4342 - val_accuracy: 0.5292 - val_loss: 1.5458 -
learning_rate: 3.0000e-05
Epoch 16/50
40/40            41s 1s/step -
accuracy: 0.9754 - loss: 0.4171 - val_accuracy: 0.5498 - val_loss: 1.6794 -
learning_rate: 3.0000e-05
Epoch 17/50
40/40            41s 1s/step -
accuracy: 0.9846 - loss: 0.3977 - val_accuracy: 0.6895 - val_loss: 1.2662 -
learning_rate: 3.0000e-05
Epoch 18/50
40/40            41s 1s/step -
accuracy: 0.9879 - loss: 0.3834 - val_accuracy: 0.7902 - val_loss: 0.8872 -
learning_rate: 3.0000e-05
Epoch 19/50
40/40            44s 1s/step -
accuracy: 0.9905 - loss: 0.3749 - val_accuracy: 0.7852 - val_loss: 0.9120 -
learning_rate: 3.0000e-05
```

```
Epoch 20/50
40/40            43s 1s/step -
accuracy: 0.9906 - loss: 0.3687 - val_accuracy: 0.7658 - val_loss: 1.0349 -
learning_rate: 3.0000e-05
Epoch 21/50
40/40            41s 1s/step -
accuracy: 0.9943 - loss: 0.3550 - val_accuracy: 0.7834 - val_loss: 0.9810 -
learning_rate: 3.0000e-05
Epoch 22/50
40/40            41s 1s/step -
accuracy: 0.9967 - loss: 0.3462 - val_accuracy: 0.8374 - val_loss: 0.7890 -
learning_rate: 3.0000e-05
Epoch 23/50
40/40            41s 1s/step -
accuracy: 0.9972 - loss: 0.3403 - val_accuracy: 0.7905 - val_loss: 0.9173 -
learning_rate: 3.0000e-05
Epoch 24/50
40/40            41s 1s/step -
accuracy: 0.9970 - loss: 0.3397 - val_accuracy: 0.7911 - val_loss: 0.9769 -
learning_rate: 3.0000e-05
Epoch 25/50
40/40            41s 1s/step -
accuracy: 0.9973 - loss: 0.3359 - val_accuracy: 0.8332 - val_loss: 0.8206 -
learning_rate: 3.0000e-05
Epoch 26/50
40/40            41s 1s/step -
accuracy: 0.9994 - loss: 0.3279 - val_accuracy: 0.7811 - val_loss: 1.1195 -
learning_rate: 3.0000e-05
Epoch 27/50
40/40            41s 1s/step -
accuracy: 0.9993 - loss: 0.3238 - val_accuracy: 0.8556 - val_loss: 0.7610 -
learning_rate: 3.0000e-05
Epoch 28/50
40/40            41s 1s/step -
accuracy: 0.9995 - loss: 0.3206 - val_accuracy: 0.8424 - val_loss: 0.7996 -
learning_rate: 3.0000e-05
Epoch 29/50
40/40            41s 1s/step -
accuracy: 0.9996 - loss: 0.3189 - val_accuracy: 0.8807 - val_loss: 0.6810 -
learning_rate: 3.0000e-05
Epoch 30/50
40/40            41s 1s/step -
accuracy: 0.9996 - loss: 0.3165 - val_accuracy: 0.8630 - val_loss: 0.7473 -
learning_rate: 3.0000e-05
Epoch 31/50
40/40            41s 1s/step -
accuracy: 1.0000 - loss: 0.3131 - val_accuracy: 0.8739 - val_loss: 0.7122 -
learning_rate: 3.0000e-05
```

```
Epoch 32/50
40/40            41s 1s/step -
accuracy: 0.9999 - loss: 0.3110 - val_accuracy: 0.8592 - val_loss: 0.7494 -
learning_rate: 3.0000e-05
Epoch 33/50
40/40            41s 1s/step -
accuracy: 0.9994 - loss: 0.3112 - val_accuracy: 0.8748 - val_loss: 0.6955 -
learning_rate: 3.0000e-05
Epoch 34/50
40/40            41s 1s/step -
accuracy: 1.0000 - loss: 0.3085 - val_accuracy: 0.8833 - val_loss: 0.6971 -
learning_rate: 3.0000e-05
Epoch 35/50
40/40            41s 1s/step -
accuracy: 1.0000 - loss: 0.3064 - val_accuracy: 0.8845 - val_loss: 0.6864 -
learning_rate: 9.0000e-06
Epoch 36/50
40/40            41s 1s/step -
accuracy: 0.9999 - loss: 0.3063 - val_accuracy: 0.8869 - val_loss: 0.6777 -
learning_rate: 9.0000e-06
Epoch 37/50
40/40            41s 1s/step -
accuracy: 0.9999 - loss: 0.3056 - val_accuracy: 0.8883 - val_loss: 0.6604 -
learning_rate: 9.0000e-06
Epoch 38/50
40/40            41s 1s/step -
accuracy: 1.0000 - loss: 0.3052 - val_accuracy: 0.8827 - val_loss: 0.6864 -
learning_rate: 9.0000e-06
Epoch 39/50
40/40            41s 1s/step -
accuracy: 1.0000 - loss: 0.3048 - val_accuracy: 0.8883 - val_loss: 0.6733 -
learning_rate: 9.0000e-06
Epoch 40/50
40/40            41s 1s/step -
accuracy: 1.0000 - loss: 0.3044 - val_accuracy: 0.8857 - val_loss: 0.6669 -
learning_rate: 9.0000e-06
Epoch 41/50
40/40            41s 1s/step -
accuracy: 1.0000 - loss: 0.3039 - val_accuracy: 0.8821 - val_loss: 0.6688 -
learning_rate: 9.0000e-06
Epoch 42/50
40/40            41s 1s/step -
accuracy: 1.0000 - loss: 0.3035 - val_accuracy: 0.8751 - val_loss: 0.7187 -
learning_rate: 9.0000e-06
Epoch 43/50
40/40            41s 1s/step -
accuracy: 1.0000 - loss: 0.3033 - val_accuracy: 0.8874 - val_loss: 0.6660 -
learning_rate: 2.7000e-06
```

```
Epoch 44/50
40/40             41s 1s/step -
accuracy: 1.0000 - loss: 0.3031 - val_accuracy: 0.8872 - val_loss: 0.6761 -
learning_rate: 2.7000e-06
Epoch 45/50
40/40             41s 1s/step -
accuracy: 1.0000 - loss: 0.3029 - val_accuracy: 0.8860 - val_loss: 0.6748 -
learning_rate: 2.7000e-06
Epoch 46/50
40/40             41s 1s/step -
accuracy: 1.0000 - loss: 0.3027 - val_accuracy: 0.8863 - val_loss: 0.6683 -
learning_rate: 2.7000e-06
Epoch 47/50
40/40             41s 1s/step -
accuracy: 1.0000 - loss: 0.3029 - val_accuracy: 0.8877 - val_loss: 0.6663 -
learning_rate: 2.7000e-06
```

[9]:
```
test_loss, test_acc = model.evaluate(test_ds)
print(f"Test accuracy: {test_acc:.3f}")
```

```
14/14             21s 2s/step -
accuracy: 0.8957 - loss: 0.6293
Test accuracy: 0.896
```

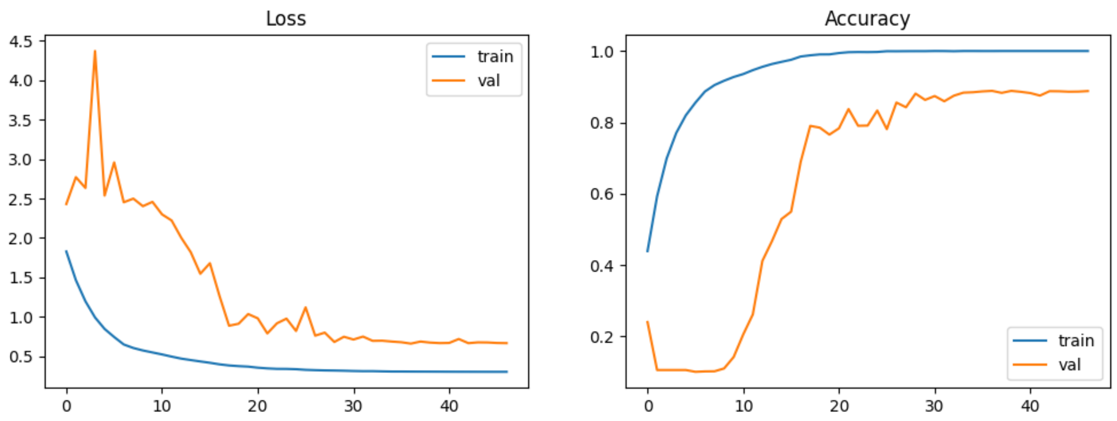
```
14/14             21s 2s/step -
accuracy: 0.8957 - loss: 0.6293
Test accuracy: 0.896
```

[10]:
```
plt.figure(figsize=(12,4))

plt.subplot(1,2,1)
plt.plot(history.history["loss"], label="train")
plt.plot(history.history["val_loss"], label="val")
plt.title("Loss")
plt.legend()

plt.subplot(1,2,2)
plt.plot(history.history["accuracy"], label="train")
plt.plot(history.history["val_accuracy"], label="val")
plt.title("Accuracy")
plt.legend()

plt.show()
```



```
[11]: model.save("advanced_model_no_data_augmentation.keras")
```