

---

# **swiss\_army\_keras**

***Release 0.3.10***

**Matteo Ferrabone**

**Feb 21, 2023**



**CONTENTS:**

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Features . . . . .	1
<b>2</b>	<b>Examples</b>	<b>3</b>
2.1	Classification Example . . . . .	3
2.2	Example segmentation with DeepLabv3+ . . . . .	35
2.3	Model User Guide . . . . .	62
<b>3</b>	<b>Indices and tables</b>	<b>73</b>





## INTRODUCTION

A library to help with the development of AI models with Keras, with a focus on edge / IoT applications.

### 1.1 Motivation

Building CNN experiments with flexible and scalable augmentation pipelines, with the ability to combine different encoder and decoder architectures, can require a huge initial effort. This is even more true in the case of edge-AI models, where models need to be built taking into account more severe constraints and quantization is usually required.

This library wants you to focus on dataset, model architecture and hyperparameters tuning, without worrying about the rest.

It provides several helper classes which help in the development of CNN based AI models for edge IoT applications, where resources are limited and model quantization is recommended.

### 1.2 Features

The main features of the library are the following:

1. Flexible, efficient and scalable Dataset management with augmentation pipelines leveraging albumentations (<https://albumentations.ai/>) and td-data (<https://www.tensorflow.org/guide/data>)
2. Training driver with builtin callbacks, configurable backbone unfreezing, and quantized model generation
3. Helper classes to easily combine pretrained backbones for Edge AI applications with the desired segmentation and classification architectures
4. Additional loss functions and optimizers which are not part of the Keras distribution, as for now



## EXAMPLES

You can find here sample notebooks which show the usage of various feature of the library in a production like experiment and setup.

1. Classification example: a classification experiment based on distiller classifier layer, with MobilenetV3Plus minimalistic feature extractor backbone.
2. Segmentation example: a segmentation experiment based on DeepLabv3+ segmentation decoder with MobilenetV3Plus minimalistic encoder.
3. Model User Guide. A notebook with documentation about the different possible models provided by this library

### 2.1 Classification Example

```
[1]: import os
import sys
from base64 import b64encode

import pathlib

import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt
from functools import partial
import albumentations as albu
```

Matplotlib created a temporary config/cache directory at /tmp/matplotlib-14mjotfm, because the default path (/usr/local/lib/python3.8/site-packages/matplotlib/.cache) is not a writable directory; it is highly recommended to set the MPLCONFIGDIR environment variable to a writable directory, in particular to speed up the import of Matplotlib and to better support multiprocessing.

Require 24 GB of GPU RAM

```
[2]: gpus = tf.config.experimental.list_physical_devices("GPU")
#os.environ['TF_GPU_ALLOCATOR'] = 'cuda_malloc_async'

if gpus:
    try:
        tf.config.experimental.set_virtual_device_configuration(
            gpus[0],
```

(continues on next page)

(continued from previous page)

```

        [
            tf.config.experimental.VirtualDeviceConfiguration(
                memory_limit=1024 * 24
            )
        ],
    )
except RuntimeError as e:
    print(e)

```

Download and setup dataset

```

[3]: flowers_root = tf.keras.utils.get_file(
    'flower_photos',
    'https://storage.googleapis.com/download.tensorflow.org/example_images/flower_photos.
    ↪tgz',
    untar=True)
flowers_root = pathlib.Path(flowers_root)

```

Import swiss\_army\_keras lib

```

[4]: #Uncomment to install dependencies, if needed
#!pip3 install --upgrade git+https://github.com/waterviewsrl/swiss-army-keras.git
#!pip3 install typeguard
#!pip3 install --upgrade irondomo
#!pip3 install --upgrade pillow
#!pip3 install --upgrade adabelief_tf
from swiss_army_keras.dataset_utils import ClassificationAlbumentationsDataLoader

from swiss_army_keras import models, base, utils, losses, optimizers

```

Define input tensor and build classifier model with default MobileNetV3Large minimalistic backbone

This model (distiller) defines a top classification layer which tries to go beyond the usual GlobalPooling + Dense layer, by preserving information about spatial relation of macrofeatures in the last convolutional layer output.

```

[5]: inp = tf.keras.layers.Input(shape=(512, 512, 3), batch_size=1)

model = models.distiller_classifier(inp, 5, macrofeatures_number=5, size=256)

```

```

WARNING:tensorflow:`input_shape` is undefined or non-square, or `rows` is not 224.
↪Weights for input shape (224, 224) will be loaded as the default.

```

Define augmentation pipeline (see <https://albumentations.ai/docs/>)

```

[6]: def get_training_augmentation(width=512, height=512):
    train_transform = [

        #albu.RandomRotate90(p=0.5),

        #albu.OneOf([

```

(continues on next page)

(continued from previous page)

```

#
#
#], p=1),
#albu.Resize(width, height, 2, p=1),

#albu.ShiftScaleRotate(shift_limit = 0.05, scale_limit=(0.1, 0.1),
↪ interpolation=2, rotate_limit=25, p=0.3),

albu.HorizontalFlip(p=0.5),
#albu.VerticalFlip(p=0.15),

albu.Compose([
albu.OneOf([
    albu.ISONoise(p=1),
    albu.MultiplicativeNoise(p=1),
    albu.GaussNoise(p=1),
    albu.ImageCompression(p=1),

    ],
    p=0.2,
),

#albu.Perspective(pad_mode=2, p=0.1),

albu.OneOf(
    [
        albu.CLAHE(p=1),
        albu.RandomBrightnessContrast(p=1),
        albu.Sharpen(p=1),
        albu.Emboss(p=1),
        albu.RandomGamma(p=1),

    ],
    p=0.5,
),

albu.OneOf(
    [
        albu.Blur(blur_limit=5, p=1),
        albu.MedianBlur(blur_limit=5, p=1),
        albu.MotionBlur(blur_limit=5, p=1),

    ],
    p=0.25,
),

albu.OneOf(
    [
        albu.RandomBrightnessContrast(p=1),
        albu.HueSaturationValue(p=1),
        albu.RGBShift(p=1),

    ],
    p=0.25,
),

```

(continues on next page)

(continued from previous page)

```

        albu.OneOf(
            [
                #albu.ElasticTransform(alpha=10, sigma=10 * 0.05, alpha_affine=10 * 0.03,
→ p=0.5),
                albu.GridDistortion(p=0.5),
                albu.OpticalDistortion(distort_limit=0.3, shift_limit=0.05, p=1)
            ],
            p=0.25,
        )], p=0.5),

        #DropsOnLens(maxR=40, minR=30, p=0.1),
        albu.ToGray(p=0.2)
    ]

    return albu.Compose(train_transform)
#return albu.Compose([])

```

Initialize classification dataloader

This will create a number of workers equal to half the number of CPU cores on the machine. These workers will produce random augmentations according to the defined pipeline, so that at each epoch every batch will change.

Please note: Passing `normalization=model.preprocessing` grants that the correct normalization is applied in a consistent way during training, evaluation and quantization.

Please note: This will produce a lot of output, depending also on the number of CPUs, to monitor workers connectivity and liveness

```

[7]: dl = ClassificationAlbumentationsDataLoader(str(flowers_root), width=512, height=512,
→ train_augmentations=get_training_augmentation(), batch_size=16, normalization=model.
→ preprocessing)

```

```
WARNING:root:Normalizing in range: [ 0 255]
```

```

Class label daisy will have id: 0
Class label dandelion will have id: 1
Class label roses will have id: 2
Class label sunflowers will have id: 3
Class label tulips will have id: 4
train_val_test_split: [2936, 367, 367]

```

```

WARNING:root:KEYS: None
WARNING:root:Router on "ipc:///tmp/swiss_army_keras_classification_clear_socket_QbuLn4Q="
WARNING:root:KEYS: None
WARNING:root:KEYS: (b'P+S690P{iVPfx<aFJwxfsY^ugFzjuW0naIh!o7J<', b':$80ST.hxA5xL7c+@
→ $3YTEohOR^GhrJ2$qzN@bR^')
WARNING:root:KEYS: None
WARNING:root:Attaching HOST CURVE keys.
WARNING:root:Dealer on "ipc:///tmp/swiss_army_keras_classification_clear_socket_QbuLn4Q="
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
→ QbuLn4Q=): {'event': 128, 'value': 96, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→ classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_CLOSED'}
WARNING:root:Router on "ipc:///tmp/swiss_army_keras_classification_curve_socket_QbuLn4Q="

```

(continues on next page)

(continued from previous page)

```

WARNING:root:Dealer on "ipc:///tmp/swiss_army_keras_classification_clear_socket_QbuLn4Q="
WARNING:root:KEYS: None
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
↳QbuLn4Q=): {'event': 4, 'value': 297, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↳classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_CONNECT_RETRIED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
↳QbuLn4Q=): {'event': 128, 'value': 97, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↳classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_CLOSED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
↳QbuLn4Q=): {'event': 8, 'value': 107, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↳classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_LISTENING'}
WARNING:root:Dealer on "ipc:///tmp/swiss_army_keras_classification_clear_socket_QbuLn4Q="
WARNING:root:KEYS: None
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
↳QbuLn4Q=): {'event': 4, 'value': 297, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↳classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_CONNECT_RETRIED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_curve_socket_
↳QbuLn4Q=): {'event': 8, 'value': 108, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↳classification_curve_socket_QbuLn4Q=', 'description': 'EVENT_LISTENING'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
↳QbuLn4Q=): {'event': 1, 'value': 98, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↳classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_CONNECTED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
↳QbuLn4Q=): {'event': 32, 'value': 109, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↳classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_ACCEPTED'}
WARNING:root:Dealer on "ipc:///tmp/swiss_army_keras_classification_clear_socket_QbuLn4Q="
WARNING:root:KEYS: None
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
↳QbuLn4Q=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↳classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
↳QbuLn4Q=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↳classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
↳QbuLn4Q=): {'event': 1, 'value': 99, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↳classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_CONNECTED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
↳QbuLn4Q=): {'event': 32, 'value': 115, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↳classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_ACCEPTED'}
WARNING:root:Dealer on "ipc:///tmp/swiss_army_keras_classification_clear_socket_QbuLn4Q="
WARNING:root:KEYS: None
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
↳QbuLn4Q=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↳classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
↳QbuLn4Q=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↳classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
↳QbuLn4Q=): {'event': 1, 'value': 100, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↳classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_CONNECTED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
↳QbuLn4Q=): {'event': 32, 'value': 116, 'endpoint': b'ipc:///tmp/swiss_army_keras_

```

(continues on next page)

(continued from previous page)

```

↪classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_ACCEPTED'}
WARNING:root:Dealer on "ipc:///tmp/swiss_army_keras_classification_clear_socket_QbuLn4Q="
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
↪QbuLn4Q=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↪classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:KEYS: None
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
↪QbuLn4Q=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↪classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
↪QbuLn4Q=): {'event': 1, 'value': 101, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↪classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_CONNECTED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
↪QbuLn4Q=): {'event': 32, 'value': 112, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↪classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_ACCEPTED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
↪QbuLn4Q=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↪classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Dealer on "ipc:///tmp/swiss_army_keras_classification_clear_socket_QbuLn4Q="
WARNING:root:KEYS: None
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
↪QbuLn4Q=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↪classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
↪QbuLn4Q=): {'event': 32, 'value': 121, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↪classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_ACCEPTED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
↪QbuLn4Q=): {'event': 1, 'value': 103, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↪classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_CONNECTED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
↪QbuLn4Q=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↪classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Dealer on "ipc:///tmp/swiss_army_keras_classification_clear_socket_QbuLn4Q="
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
↪QbuLn4Q=): {'event': 1, 'value': 96, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↪classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_CONNECTED'}
WARNING:root:KEYS: None
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
↪QbuLn4Q=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↪classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
↪QbuLn4Q=): {'event': 32, 'value': 125, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↪classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_ACCEPTED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
↪QbuLn4Q=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↪classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
↪QbuLn4Q=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↪classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:KEYS: None
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
↪QbuLn4Q=): {'event': 1, 'value': 104, 'endpoint': b'ipc:///tmp/swiss_army_keras_

```

(continues on next page)



(continued from previous page)

```

→classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_CONNECTED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
→QbuLn4Q=): {'event': 32, 'value': 126, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_ACCEPTED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
→QbuLn4Q=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
→QbuLn4Q=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Dealer on "ipc:///tmp/swiss_army_keras_classification_clear_socket_QbuLn4Q="
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
→QbuLn4Q=): {'event': 1, 'value': 97, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_CONNECTED'}
WARNING:root:Dealer on "ipc:///tmp/swiss_army_keras_classification_clear_socket_QbuLn4Q="
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
→QbuLn4Q=): {'event': 32, 'value': 127, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_ACCEPTED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
→QbuLn4Q=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
→QbuLn4Q=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
→QbuLn4Q=): {'event': 1, 'value': 105, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_CONNECTED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
→QbuLn4Q=): {'event': 1, 'value': 106, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_CONNECTED'}
WARNING:root:KEYS: None
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
→QbuLn4Q=): {'event': 32, 'value': 110, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_ACCEPTED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
→QbuLn4Q=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
→QbuLn4Q=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
→QbuLn4Q=): {'event': 32, 'value': 111, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_ACCEPTED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
→QbuLn4Q=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Dealer on "ipc:///tmp/swiss_army_keras_classification_clear_socket_QbuLn4Q="
WARNING:root:KEYS: None
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
→QbuLn4Q=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_

```

(continues on next page)

(continued from previous page)

[illegible]

(continues on next page)

(continued from previous page)

```

↪classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Dealer on "ipc:///tmp/swiss_army_keras_classification_clear_socket_QbuLn4Q="
WARNING:root:KEYS: None
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
↪QbuLn4Q=): {'event': 32, 'value': 113, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↪classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_ACCEPTED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
↪QbuLn4Q=): {'event': 1, 'value': 112, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↪classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_CONNECTED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
↪QbuLn4Q=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↪classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
↪QbuLn4Q=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↪classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Dealer on "ipc:///tmp/swiss_army_keras_classification_clear_socket_QbuLn4Q="
WARNING:root:KEYS: None
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
↪QbuLn4Q=): {'event': 32, 'value': 114, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↪classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_ACCEPTED'}
WARNING:root:KEYS: None
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
↪QbuLn4Q=): {'event': 1, 'value': 112, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↪classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_CONNECTED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
↪QbuLn4Q=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↪classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
↪QbuLn4Q=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↪classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Dealer on "ipc:///tmp/swiss_army_keras_classification_clear_socket_QbuLn4Q="
WARNING:root:Dealer on "ipc:///tmp/swiss_army_keras_classification_clear_socket_QbuLn4Q="
WARNING:root:KEYS: None
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
↪QbuLn4Q=): {'event': 32, 'value': 117, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↪classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_ACCEPTED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
↪QbuLn4Q=): {'event': 1, 'value': 112, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↪classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_CONNECTED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
↪QbuLn4Q=): {'event': 1, 'value': 114, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↪classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_CONNECTED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
↪QbuLn4Q=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↪classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
↪QbuLn4Q=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↪classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Dealer on "ipc:///tmp/swiss_army_keras_classification_clear_socket_QbuLn4Q="
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
↪QbuLn4Q=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↪classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}

```

(continues on next page)

(continued from previous page)

```

WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
↳QbuLn4Q=): {'event': 32, 'value': 137, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↳classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_ACCEPTED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
↳QbuLn4Q=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↳classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
↳QbuLn4Q=): {'event': 1, 'value': 113, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↳classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_CONNECTED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
↳QbuLn4Q=): {'event': 32, 'value': 134, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↳classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_ACCEPTED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
↳QbuLn4Q=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↳classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_classification_clear_socket_
↳QbuLn4Q=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↳classification_clear_socket_QbuLn4Q=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}

```

Define train val and test sets

```
[8]: train_set, val_set, test_set = dl.build_datasets()
```

View a batch of augmented images from train set

```
[9]: dl.show_images(16)
```



Image



Image





Image

dandelion



Image

daisy



Image



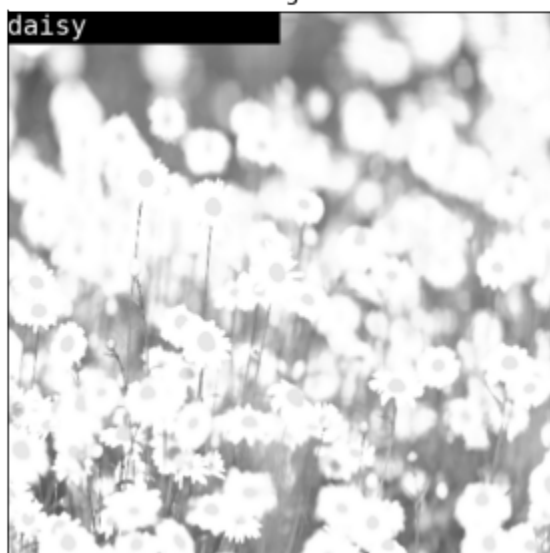
Image



Image



Image





Image

dandelion



Image

dandelion



Image



Image



Image

dandelion



Image

roses





```
[10]: from adabelief_tf import AdaBeliefOptimizer
      from swiss_army_keras.training_utils import TrainingDriver
```

```
[11]: accuracy = tf.keras.metrics.CategoricalAccuracy()
```

Train for 40 epochs (20 with frozen backend + 20 with unfrozen backend) with the desired optimizer, loss and metrics.

This will also produce h5 model, f16 and ui8 quantized models, metrics and loss history graphs, all named according to the name (prova\_multi in this case) + startup datetime.

Please Note: TrainingDriver implicitly defines callbacks for ModelCheckpoint and for EarlyStopping with 15 epochs patience. The saved models, quantized and not, corresponds to the best validation loss found during training.

```
[12]: td = TrainingDriver(model, 'prova_multi', AdaBeliefOptimizer(learning_rate=1e-4,
    ↪epsilon=1e-14, rectify=True), tf.keras.losses.CategoricalCrossentropy(label_
    ↪smoothing=0.1), [accuracy], train_set, val_set, test_set, 20 )
      td.run()
```

Please check your arguments if you have upgraded adabelief-tf from version 0.0.1.  
Modifications to default arguments:

	eps	weight_decouple	rectify
adabelief-tf=0.0.1	1e-08	Not supported	Not supported
>=0.1.0 (Current 0.2.1)	1e-14	supported	default: True

SGD better than Adam (e.g. CNN for Image Classification)      Adam better than SGD (e.g. Transformer, GAN)

Recommended epsilon = 1e-7      Recommended epsilon = 1e-14  
For a complete table of recommended hyperparameters, see  
<https://github.com/juntang-zhuang/Adabelief-Optimizer>  
You can disable the log message by setting "print\_change\_log = False", though it is recommended to keep as a reminder.

(continues on next page)

(continued from previous page)

Model: "model"

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(1, 512, 512, 3)]	0	
MobileNetV3Large_backbone (Func	[(1, 256, 256, 64),	1437608	input_1[0][0]
average_pooling2d (AveragePooli	(1, 5, 5, 960)	0	MobileNetV3Large_ backbone[0][4]
depthwise_conv2d (DepthwiseConv	(1, 1, 1, 960)	24960	average_pooling2d[0][0]
depthwise_conv2d_1 (DepthwiseCo	(1, 1, 1, 960)	24960	average_pooling2d[0][0]
depthwise_conv2d_2 (DepthwiseCo	(1, 1, 1, 960)	24960	average_pooling2d[0][0]
depthwise_conv2d_3 (DepthwiseCo	(1, 1, 1, 960)	24960	average_pooling2d[0][0]
depthwise_conv2d_4 (DepthwiseCo	(1, 1, 1, 960)	24960	average_pooling2d[0][0]
flatten (Flatten)	(1, 960)	0	depthwise_conv2d[0][0]
flatten_1 (Flatten)	(1, 960)	0	depthwise_conv2d_1[0][0]
flatten_2 (Flatten)	(1, 960)	0	depthwise_conv2d_2[0][0]
flatten_3 (Flatten)	(1, 960)	0	depthwise_conv2d_3[0][0]
flatten_4 (Flatten)	(1, 960)	0	depthwise_conv2d_4[0][0]
dropout (Dropout)	(1, 960)	0	flatten[0][0]
dropout_1 (Dropout)	(1, 960)	0	flatten_1[0][0]

(continues on next page)

(continued from previous page)

dropout_2 (Dropout)	(1, 960)	0	flatten_2[0][0]
↪-----			
dropout_3 (Dropout)	(1, 960)	0	flatten_3[0][0]
↪-----			
dropout_4 (Dropout)	(1, 960)	0	flatten_4[0][0]
↪-----			
concatenate (Concatenate)	(1, 4800)	0	dropout[0][0] dropout_1[0][0] dropout_2[0][0] dropout_3[0][0] dropout_4[0][0]
↪-----			
dense (Dense)	(1, 256)	1229056	concatenate[0][0]
↪-----			
dropout_5 (Dropout)	(1, 256)	0	dense[0][0]
↪-----			
dense_1 (Dense)	(1, 5)	1285	dropout_5[0][0]
=====			
Total params: 2,792,749			
Trainable params: 1,355,141			
Non-trainable params: 1,437,608			
=====			
↪-----			
Epoch 1/20			
WARNING:tensorflow:Model was constructed with shape (1, 512, 512, 3) for input_			
↪KerasTensor(type_spec=TensorSpec(shape=(1, 512, 512, 3), dtype=tf.float32, name='input_			
↪1'), name='input_1', description="created by layer 'input_1'"), but it was called on_			
↪an input with incompatible shape (16, 512, 512, 3).			
/usr/local/lib/python3.6/dist-packages/keras/utils/generic_utils.py:497:			
↪CustomMaskWarning: Custom mask layers require a config and must override get_config_			
↪When loading, the custom mask layer must be passed to the custom_objects argument.			
category=CustomMaskWarning)			
WARNING:tensorflow:Model was constructed with shape (1, 512, 512, 3) for input_			
↪KerasTensor(type_spec=TensorSpec(shape=(1, 512, 512, 3), dtype=tf.float32, name='input_			
↪1'), name='input_1', description="created by layer 'input_1'"), but it was called on_			
↪an input with incompatible shape (16, 512, 512, 3).			
WARNING:tensorflow:Model was constructed with shape (1, 512, 512, 3) for input_			
↪KerasTensor(type_spec=TensorSpec(shape=(1, 512, 512, 3), dtype=tf.float32, name='input_			
↪1'), name='input_1', description="created by layer 'input_1'"), but it was called on_			
↪an input with incompatible shape (16, 512, 512, 3).			
WARNING:tensorflow:Model was constructed with shape (1, 512, 512, 3) for input_			
↪KerasTensor(type_spec=TensorSpec(shape=(1, 512, 512, 3), dtype=tf.float32, name='input_			
↪1'), name='input_1', description="created by layer 'input_1'"), but it was called on_			
↪an input with incompatible shape (16, 512, 512, 3).			

```

WARNING:tensorflow:Model was constructed with shape (1, 512, 512, 3) for input_
↳KerasTensor(type_spec=TensorSpec(shape=(1, 512, 512, 3), dtype=tf.float32, name='input_
↳1'), name='input_1', description="created by layer 'input_1'"), but it was called on_
↳an input with incompatible shape (16, 512, 512, 3).

WARNING:tensorflow:Model was constructed with shape (1, 512, 512, 3) for input_
↳KerasTensor(type_spec=TensorSpec(shape=(1, 512, 512, 3), dtype=tf.float32, name='input_
↳1'), name='input_1', description="created by layer 'input_1'"), but it was called on_
↳an input with incompatible shape (16, 512, 512, 3).

WARNING:tensorflow:Model was constructed with shape (1, 512, 512, 3) for input_
↳KerasTensor(type_spec=TensorSpec(shape=(1, 512, 512, 3), dtype=tf.float32, name='input_
↳1'), name='input_1', description="created by layer 'input_1'"), but it was called on_
↳an input with incompatible shape (16, 512, 512, 3).

WARNING:tensorflow:Model was constructed with shape (1, 512, 512, 3) for input_
↳KerasTensor(type_spec=TensorSpec(shape=(1, 512, 512, 3), dtype=tf.float32, name='input_
↳1'), name='input_1', description="created by layer 'input_1'"), but it was called on_
↳an input with incompatible shape (16, 512, 512, 3).

183/183 [=====] - ETA: 0s - loss: 2.0143 - categorical_accuracy:
↳ 0.5249WARNING:tensorflow:Model was constructed with shape (1, 512, 512, 3) for input_
↳KerasTensor(type_spec=TensorSpec(shape=(1, 512, 512, 3), dtype=tf.float32, name='input_
↳1'), name='input_1', description="created by layer 'input_1'"), but it was called on_
↳an input with incompatible shape (16, 512, 512, 3).

WARNING:tensorflow:Model was constructed with shape (1, 512, 512, 3) for input_
↳KerasTensor(type_spec=TensorSpec(shape=(1, 512, 512, 3), dtype=tf.float32, name='input_
↳1'), name='input_1', description="created by layer 'input_1'"), but it was called on_
↳an input with incompatible shape (16, 512, 512, 3).

WARNING:tensorflow:Model was constructed with shape (1, 512, 512, 3) for input_
↳KerasTensor(type_spec=TensorSpec(shape=(1, 512, 512, 3), dtype=tf.float32, name='input_
↳1'), name='input_1', description="created by layer 'input_1'"), but it was called on_
↳an input with incompatible shape (16, 512, 512, 3).

WARNING:tensorflow:Model was constructed with shape (1, 512, 512, 3) for input_
↳KerasTensor(type_spec=TensorSpec(shape=(1, 512, 512, 3), dtype=tf.float32, name='input_
↳1'), name='input_1', description="created by layer 'input_1'"), but it was called on_
↳an input with incompatible shape (16, 512, 512, 3).

183/183 [=====] - 16s 55ms/step - loss: 2.0143 - categorical_
↳accuracy: 0.5249 - val_loss: 1.8842 - val_categorical_accuracy: 0.6591
Epoch 2/20
183/183 [=====] - 10s 50ms/step - loss: 1.6382 - categorical_
↳accuracy: 0.7510 - val_loss: 1.4155 - val_categorical_accuracy: 0.8040
Epoch 3/20
183/183 [=====] - 10s 52ms/step - loss: 1.2311 - categorical_
↳accuracy: 0.8463 - val_loss: 1.1303 - val_categorical_accuracy: 0.8438
Epoch 4/20
183/183 [=====] - 10s 51ms/step - loss: 1.0534 - categorical_
↳accuracy: 0.8736 - val_loss: 1.0248 - val_categorical_accuracy: 0.8551
Epoch 5/20
183/183 [=====] - 10s 54ms/step - loss: 0.9722 - categorical_
↳accuracy: 0.8863 - val_loss: 0.9528 - val_categorical_accuracy: 0.8693
Epoch 6/20
183/183 [=====] - 10s 53ms/step - loss: 0.9048 - categorical_

```

(continues on next page)



(continued from previous page)

```

↳ accuracy: 0.8975 - val_loss: 0.8957 - val_categorical_accuracy: 0.8807
Epoch 7/20
183/183 [=====] - 10s 51ms/step - loss: 0.8534 - categorical_
↳ accuracy: 0.9088 - val_loss: 0.8701 - val_categorical_accuracy: 0.8778
Epoch 8/20
183/183 [=====] - 10s 51ms/step - loss: 0.8143 - categorical_
↳ accuracy: 0.9208 - val_loss: 0.8423 - val_categorical_accuracy: 0.8778
Epoch 9/20
183/183 [=====] - 10s 53ms/step - loss: 0.7829 - categorical_
↳ accuracy: 0.9255 - val_loss: 0.8147 - val_categorical_accuracy: 0.8864
Epoch 10/20
183/183 [=====] - 10s 53ms/step - loss: 0.7551 - categorical_
↳ accuracy: 0.9324 - val_loss: 0.7910 - val_categorical_accuracy: 0.8977
Epoch 11/20
183/183 [=====] - 10s 53ms/step - loss: 0.7279 - categorical_
↳ accuracy: 0.9372 - val_loss: 0.7781 - val_categorical_accuracy: 0.8977
Epoch 12/20
183/183 [=====] - 10s 53ms/step - loss: 0.7150 - categorical_
↳ accuracy: 0.9430 - val_loss: 0.7659 - val_categorical_accuracy: 0.9006
Epoch 13/20
183/183 [=====] - 10s 53ms/step - loss: 0.6990 - categorical_
↳ accuracy: 0.9406 - val_loss: 0.7563 - val_categorical_accuracy: 0.9006
Epoch 14/20
183/183 [=====] - 10s 54ms/step - loss: 0.6879 - categorical_
↳ accuracy: 0.9423 - val_loss: 0.7415 - val_categorical_accuracy: 0.8977
Epoch 15/20
183/183 [=====] - 10s 52ms/step - loss: 0.6765 - categorical_
↳ accuracy: 0.9491 - val_loss: 0.7342 - val_categorical_accuracy: 0.8977
Epoch 16/20
183/183 [=====] - 10s 52ms/step - loss: 0.6601 - categorical_
↳ accuracy: 0.9529 - val_loss: 0.7322 - val_categorical_accuracy: 0.8977
Epoch 17/20
183/183 [=====] - 10s 52ms/step - loss: 0.6540 - categorical_
↳ accuracy: 0.9532 - val_loss: 0.7276 - val_categorical_accuracy: 0.8920
Epoch 18/20
183/183 [=====] - 10s 54ms/step - loss: 0.6433 - categorical_
↳ accuracy: 0.9583 - val_loss: 0.7173 - val_categorical_accuracy: 0.9091
Epoch 19/20
183/183 [=====] - 10s 52ms/step - loss: 0.6379 - categorical_
↳ accuracy: 0.9577 - val_loss: 0.7139 - val_categorical_accuracy: 0.9034
Epoch 20/20
183/183 [=====] - 9s 50ms/step - loss: 0.6310 - categorical_
↳ accuracy: 0.9604 - val_loss: 0.7095 - val_categorical_accuracy: 0.9062

```

WARNING:root:Unfreezing Model

Model: "model"

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(1, 512, 512, 3)]	0	

(continues on next page)



(continued from previous page)

↪ _____ MobileNetV3Large_backbone (Func [(1, 256, 256, 64), 1437608		input_1[0][0]	
↪ _____ average_pooling2d (AveragePooli (1, 5, 5, 960)	0	MobileNetV3Large_ ↪ backbone[0][4]	
↪ _____ depthwise_conv2d (DepthwiseConv (1, 1, 1, 960)	24960	average_pooling2d[0][0]	
↪ _____ depthwise_conv2d_1 (DepthwiseCo (1, 1, 1, 960)	24960	average_pooling2d[0][0]	
↪ _____ depthwise_conv2d_2 (DepthwiseCo (1, 1, 1, 960)	24960	average_pooling2d[0][0]	
↪ _____ depthwise_conv2d_3 (DepthwiseCo (1, 1, 1, 960)	24960	average_pooling2d[0][0]	
↪ _____ depthwise_conv2d_4 (DepthwiseCo (1, 1, 1, 960)	24960	average_pooling2d[0][0]	
↪ _____ flatten (Flatten)	(1, 960)	0	depthwise_conv2d[0][0]
↪ _____ flatten_1 (Flatten)	(1, 960)	0	depthwise_conv2d_1[0][0]
↪ _____ flatten_2 (Flatten)	(1, 960)	0	depthwise_conv2d_2[0][0]
↪ _____ flatten_3 (Flatten)	(1, 960)	0	depthwise_conv2d_3[0][0]
↪ _____ flatten_4 (Flatten)	(1, 960)	0	depthwise_conv2d_4[0][0]
↪ _____ dropout (Dropout)	(1, 960)	0	flatten[0][0]
↪ _____ dropout_1 (Dropout)	(1, 960)	0	flatten_1[0][0]
↪ _____ dropout_2 (Dropout)	(1, 960)	0	flatten_2[0][0]
↪ _____ dropout_3 (Dropout)	(1, 960)	0	flatten_3[0][0]
↪ _____ dropout_4 (Dropout)	(1, 960)	0	flatten_4[0][0]

(continues on next page)

(continued from previous page)

```

→ -----
concatenate (Concatenate)      (1, 4800)      0      dropout[0][0]
                                dropout_1[0][0]
                                dropout_2[0][0]
                                dropout_3[0][0]
                                dropout_4[0][0]
-----
→ -----
dense (Dense)                  (1, 256)      1229056    concatenate[0][0]
-----
→ -----
dropout_5 (Dropout)           (1, 256)      0      dense[0][0]
-----
→ -----
dense_1 (Dense)               (1, 5)        1285      dropout_5[0][0]
=====
Total params: 2,792,749
Trainable params: 2,768,349
Non-trainable params: 24,400
-----
→ -----
Epoch 1/20
WARNING:tensorflow:Model was constructed with shape (1, 512, 512, 3) for input_
→ KerasTensor(type_spec=TensorSpec(shape=(1, 512, 512, 3), dtype=tf.float32, name='input_
→ 1'), name='input_1', description="created by layer 'input_1'"), but it was called on_
→ an input with incompatible shape (16, 512, 512, 3).

WARNING:tensorflow:Model was constructed with shape (1, 512, 512, 3) for input_
→ KerasTensor(type_spec=TensorSpec(shape=(1, 512, 512, 3), dtype=tf.float32, name='input_
→ 1'), name='input_1', description="created by layer 'input_1'"), but it was called on_
→ an input with incompatible shape (16, 512, 512, 3).

WARNING:tensorflow:Model was constructed with shape (1, 512, 512, 3) for input_
→ KerasTensor(type_spec=TensorSpec(shape=(1, 512, 512, 3), dtype=tf.float32, name='input_
→ 1'), name='input_1', description="created by layer 'input_1'"), but it was called on_
→ an input with incompatible shape (16, 512, 512, 3).

WARNING:tensorflow:Model was constructed with shape (1, 512, 512, 3) for input_
→ KerasTensor(type_spec=TensorSpec(shape=(1, 512, 512, 3), dtype=tf.float32, name='input_
→ 1'), name='input_1', description="created by layer 'input_1'"), but it was called on_
→ an input with incompatible shape (16, 512, 512, 3).

WARNING:tensorflow:Model was constructed with shape (1, 512, 512, 3) for input_
→ KerasTensor(type_spec=TensorSpec(shape=(1, 512, 512, 3), dtype=tf.float32, name='input_
→ 1'), name='input_1', description="created by layer 'input_1'"), but it was called on_
→ an input with incompatible shape (16, 512, 512, 3).

WARNING:tensorflow:Model was constructed with shape (1, 512, 512, 3) for input_
→ KerasTensor(type_spec=TensorSpec(shape=(1, 512, 512, 3), dtype=tf.float32, name='input_
→ 1'), name='input_1', description="created by layer 'input_1'"), but it was called on_
→ an input with incompatible shape (16, 512, 512, 3).

```

(continues on next page)

(continued from previous page)

↪an input with incompatible shape (16, 512, 512, 3).

WARNING:tensorflow:Model was constructed with shape (1, 512, 512, 3) for input\_1  
↪KerasTensor(type\_spec=TensorSpec(shape=(1, 512, 512, 3), dtype=tf.float32, name='input\_1'), name='input\_1', description="created by layer 'input\_1'"), but it was called on  
↪an input with incompatible shape (16, 512, 512, 3).

183/183 [=====] - ETA: 0s - loss: 0.7263 - categorical\_accuracy:  
↪0.9095WARNING:tensorflow:Model was constructed with shape (1, 512, 512, 3) for input\_1  
↪KerasTensor(type\_spec=TensorSpec(shape=(1, 512, 512, 3), dtype=tf.float32, name='input\_1'), name='input\_1', description="created by layer 'input\_1'"), but it was called on  
↪an input with incompatible shape (16, 512, 512, 3).

WARNING:tensorflow:Model was constructed with shape (1, 512, 512, 3) for input\_1  
↪KerasTensor(type\_spec=TensorSpec(shape=(1, 512, 512, 3), dtype=tf.float32, name='input\_1'), name='input\_1', description="created by layer 'input\_1'"), but it was called on  
↪an input with incompatible shape (16, 512, 512, 3).

WARNING:tensorflow:Model was constructed with shape (1, 512, 512, 3) for input\_1  
↪KerasTensor(type\_spec=TensorSpec(shape=(1, 512, 512, 3), dtype=tf.float32, name='input\_1'), name='input\_1', description="created by layer 'input\_1'"), but it was called on  
↪an input with incompatible shape (16, 512, 512, 3).

WARNING:tensorflow:Model was constructed with shape (1, 512, 512, 3) for input\_1  
↪KerasTensor(type\_spec=TensorSpec(shape=(1, 512, 512, 3), dtype=tf.float32, name='input\_1'), name='input\_1', description="created by layer 'input\_1'"), but it was called on  
↪an input with incompatible shape (16, 512, 512, 3).

183/183 [=====] - 63s 244ms/step - loss: 0.7263 - categorical\_accuracy: 0.9095 - val\_loss: 0.8143 - val\_categorical\_accuracy: 0.8608

Epoch 2/20

183/183 [=====] - 42s 226ms/step - loss: 0.6166 - categorical\_accuracy: 0.9611 - val\_loss: 0.7439 - val\_categorical\_accuracy: 0.8807

Epoch 3/20

183/183 [=====] - 41s 223ms/step - loss: 0.5775 - categorical\_accuracy: 0.9775 - val\_loss: 0.6824 - val\_categorical\_accuracy: 0.9062

Epoch 4/20

183/183 [=====] - 41s 224ms/step - loss: 0.5598 - categorical\_accuracy: 0.9843 - val\_loss: 0.6625 - val\_categorical\_accuracy: 0.9091

Epoch 5/20

183/183 [=====] - 41s 224ms/step - loss: 0.5425 - categorical\_accuracy: 0.9870 - val\_loss: 0.6520 - val\_categorical\_accuracy: 0.9006

Epoch 6/20

183/183 [=====] - 41s 225ms/step - loss: 0.5324 - categorical\_accuracy: 0.9908 - val\_loss: 0.6757 - val\_categorical\_accuracy: 0.8892

Epoch 7/20

183/183 [=====] - 42s 228ms/step - loss: 0.5218 - categorical\_accuracy: 0.9911 - val\_loss: 0.6407 - val\_categorical\_accuracy: 0.9176

Epoch 8/20

183/183 [=====] - 41s 223ms/step - loss: 0.5139 - categorical\_accuracy: 0.9928 - val\_loss: 0.6174 - val\_categorical\_accuracy: 0.9318

Epoch 9/20

183/183 [=====] - 42s 225ms/step - loss: 0.5096 - categorical\_accuracy: 0.9915 - val\_loss: 0.6094 - val\_categorical\_accuracy: 0.9261

Epoch 10/20

(continues on next page)

(continued from previous page)

```

183/183 [=====] - 42s 227ms/step - loss: 0.4980 - categorical_
↳ accuracy: 0.9949 - val_loss: 0.6024 - val_categorical_accuracy: 0.9318
Epoch 11/20
183/183 [=====] - 42s 228ms/step - loss: 0.4962 - categorical_
↳ accuracy: 0.9928 - val_loss: 0.6056 - val_categorical_accuracy: 0.9290
Epoch 12/20
183/183 [=====] - 42s 226ms/step - loss: 0.4871 - categorical_
↳ accuracy: 0.9956 - val_loss: 0.6182 - val_categorical_accuracy: 0.9318
Epoch 13/20
183/183 [=====] - 42s 225ms/step - loss: 0.4838 - categorical_
↳ accuracy: 0.9949 - val_loss: 0.5962 - val_categorical_accuracy: 0.9318
Epoch 14/20
183/183 [=====] - 41s 223ms/step - loss: 0.4812 - categorical_
↳ accuracy: 0.9945 - val_loss: 0.5968 - val_categorical_accuracy: 0.9375
Epoch 15/20
183/183 [=====] - 41s 222ms/step - loss: 0.4762 - categorical_
↳ accuracy: 0.9959 - val_loss: 0.5996 - val_categorical_accuracy: 0.9290
Epoch 16/20
183/183 [=====] - 41s 224ms/step - loss: 0.4726 - categorical_
↳ accuracy: 0.9956 - val_loss: 0.5892 - val_categorical_accuracy: 0.9347
Epoch 17/20
183/183 [=====] - 41s 224ms/step - loss: 0.4675 - categorical_
↳ accuracy: 0.9973 - val_loss: 0.5873 - val_categorical_accuracy: 0.9318
Epoch 18/20
183/183 [=====] - 41s 224ms/step - loss: 0.4639 - categorical_
↳ accuracy: 0.9966 - val_loss: 0.5792 - val_categorical_accuracy: 0.9318
Epoch 19/20
183/183 [=====] - 42s 225ms/step - loss: 0.4599 - categorical_
↳ accuracy: 0.9980 - val_loss: 0.5845 - val_categorical_accuracy: 0.9318
Epoch 20/20
183/183 [=====] - 41s 222ms/step - loss: 0.4561 - categorical_
↳ accuracy: 0.9973 - val_loss: 0.5657 - val_categorical_accuracy: 0.9403

```

WARNING:root:Quantizing Model

WARNING:tensorflow:Compiled the loaded model, but the compiled metrics have yet to be built. `model.compile\_metrics` will be empty until you train or evaluate the model.

WARNING:tensorflow:Compiled the loaded model, but the compiled metrics have yet to be built. `model.compile\_metrics` will be empty until you train or evaluate the model.

INFO:tensorflow:Assets written to: /tmp/tmp2chah2d2/assets

INFO:tensorflow:Assets written to: /tmp/tmp2chah2d2/assets  
/usr/local/lib/python3.6/dist-packages/keras/utils/generic\_utils.py:497: CustomMaskWarning: Custom mask layers require a config and must override get\_config.  
↳ When loading, the custom mask layer must be passed to the custom\_objects argument.  
category=CustomMaskWarning)

WARNING:tensorflow:Issue encountered when serializing table\_initializer.  
Type is unsupported, or the types of the items don't match field type in CollectionDef.  
↳ Note this is a warning and probably safe to ignore.  
'NoneType' object has no attribute 'name'

WARNING:tensorflow:Issue encountered when serializing table\_initializer.  
Type is unsupported, or the types of the items don't match field type in CollectionDef.

(continues on next page)

(continued from previous page)

```

↳Note this is a warning and probably safe to ignore.
'NoneType' object has no attribute 'name'
WARNING:absl:For model inputs containing unsupported operations which cannot be
↳quantized, the `inference_input_type` attribute will default to the original type.

WARNING:tensorflow:Compiled the loaded model, but the compiled metrics have yet to be
↳built. `model.compile_metrics` will be empty until you train or evaluate the model.

WARNING:tensorflow:Compiled the loaded model, but the compiled metrics have yet to be
↳built. `model.compile_metrics` will be empty until you train or evaluate the model.

INFO:tensorflow:Assets written to: /tmp/tmpjiq88zc4/assets

INFO:tensorflow:Assets written to: /tmp/tmpjiq88zc4/assets
/usr/local/lib/python3.6/dist-packages/keras/utils/generic_utils.py:497:
↳CustomMaskWarning: Custom mask layers require a config and must override get_config.
↳When loading, the custom mask layer must be passed to the custom_objects argument.
category=CustomMaskWarning)

WARNING:tensorflow:Issue encountered when serializing table_initializer.
Type is unsupported, or the types of the items don't match field type in CollectionDef.
↳Note this is a warning and probably safe to ignore.
'NoneType' object has no attribute 'name'

WARNING:tensorflow:Issue encountered when serializing table_initializer.
Type is unsupported, or the types of the items don't match field type in CollectionDef.
↳Note this is a warning and probably safe to ignore.
'NoneType' object has no attribute 'name'

<Figure size 432x288 with 0 Axes>

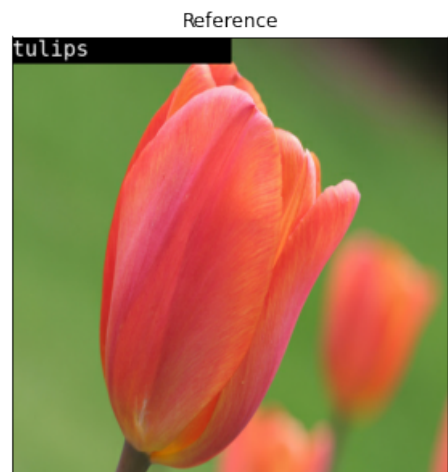
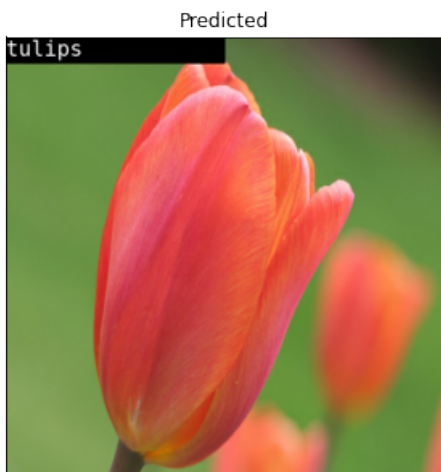
```

Show some results with corresponding reference classification

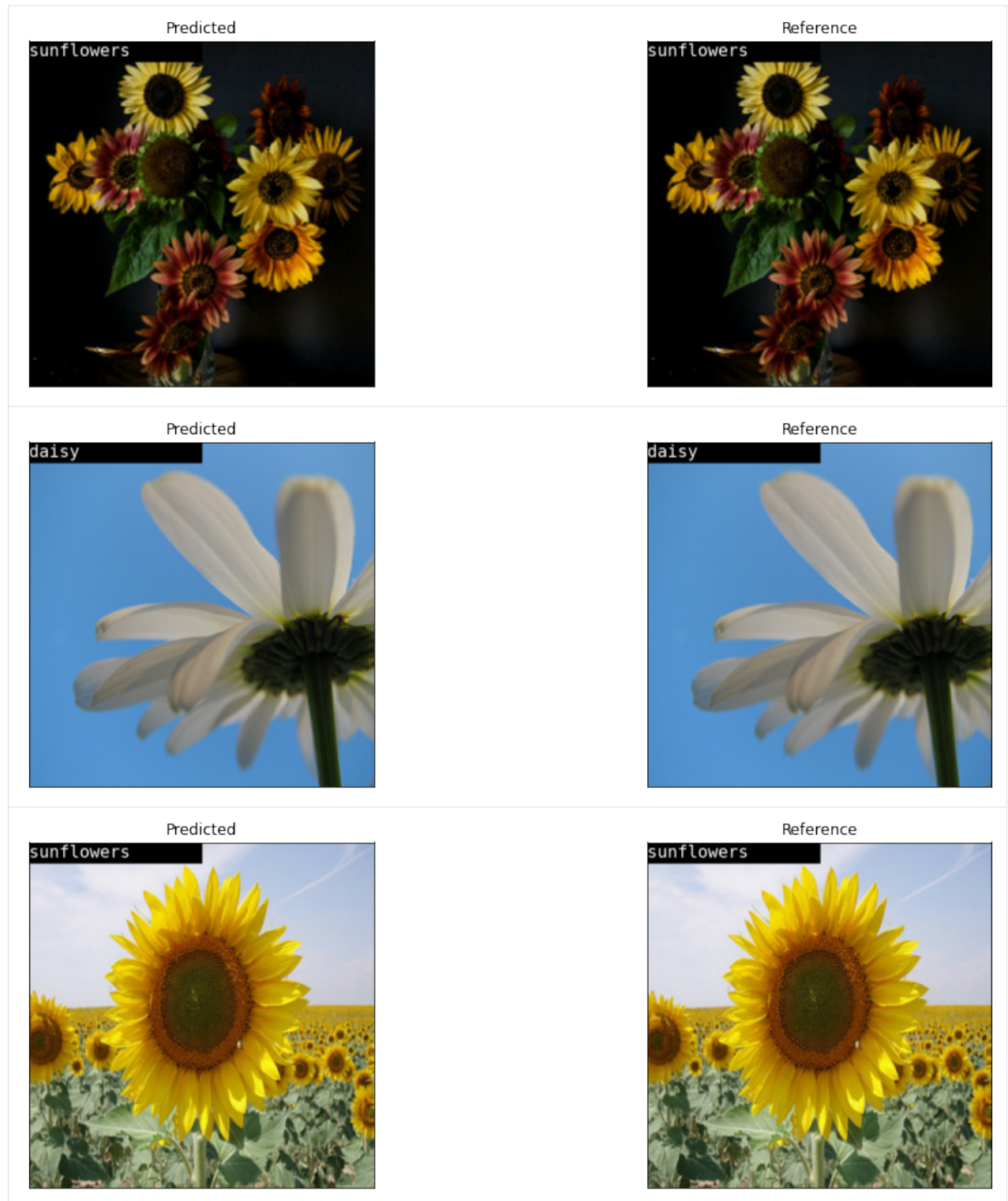
```
[13]: dl.show_results(model, 16)
```

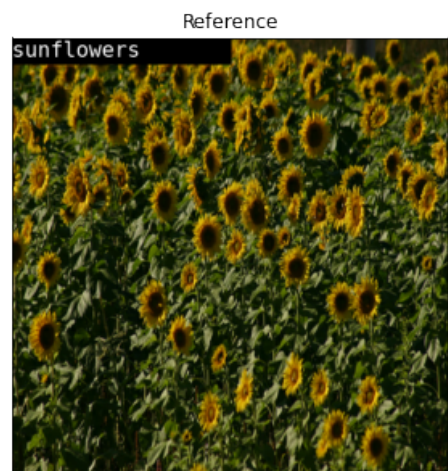
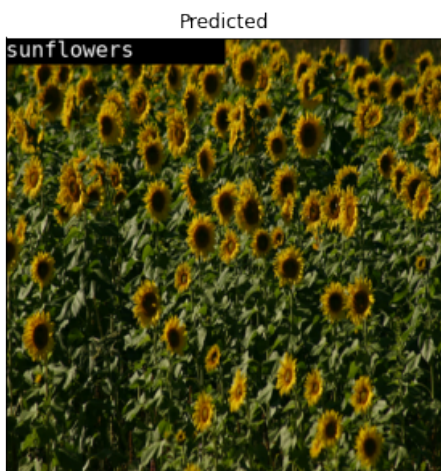
```
<Figure size 1584x1584 with 0 Axes>
```



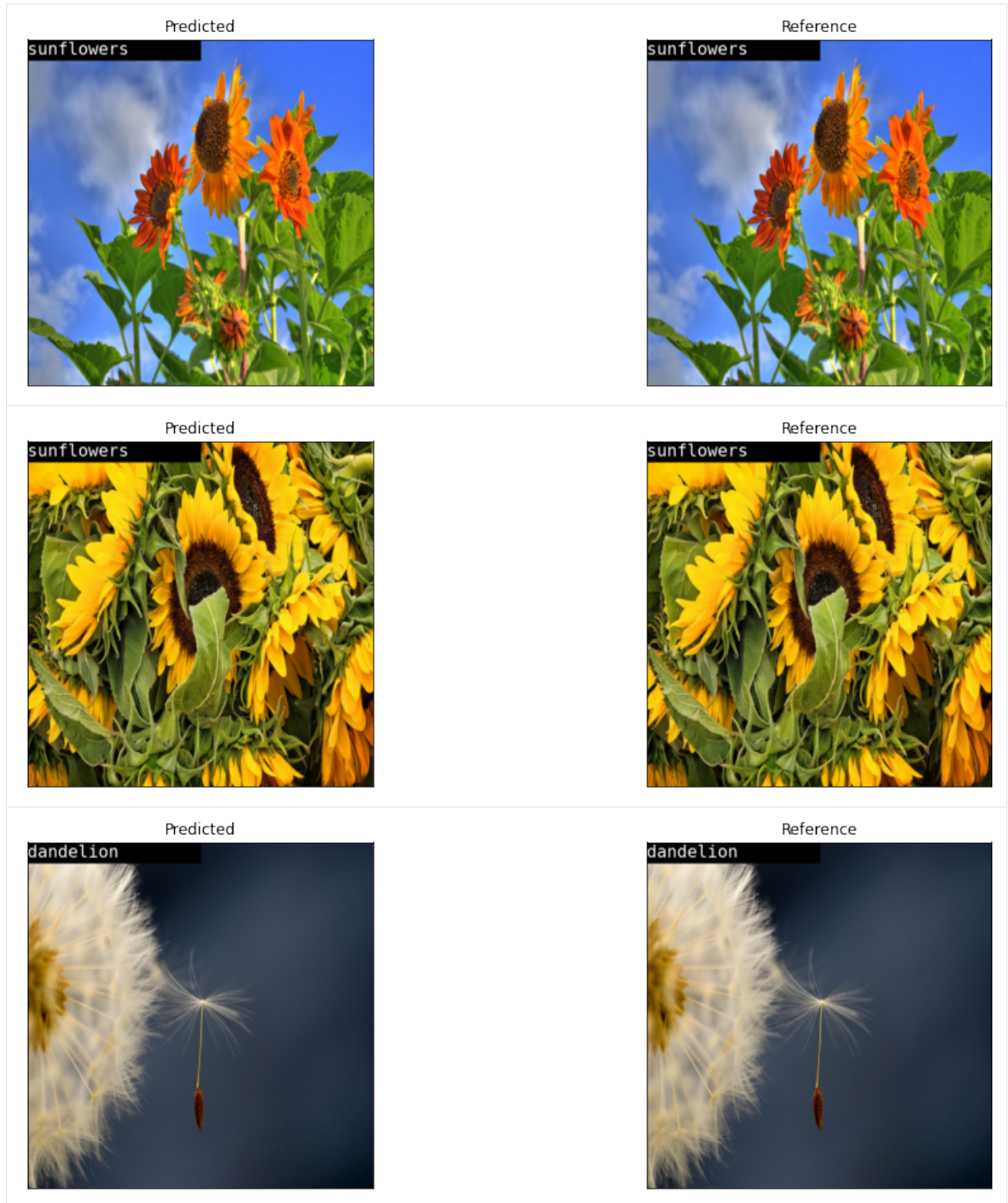


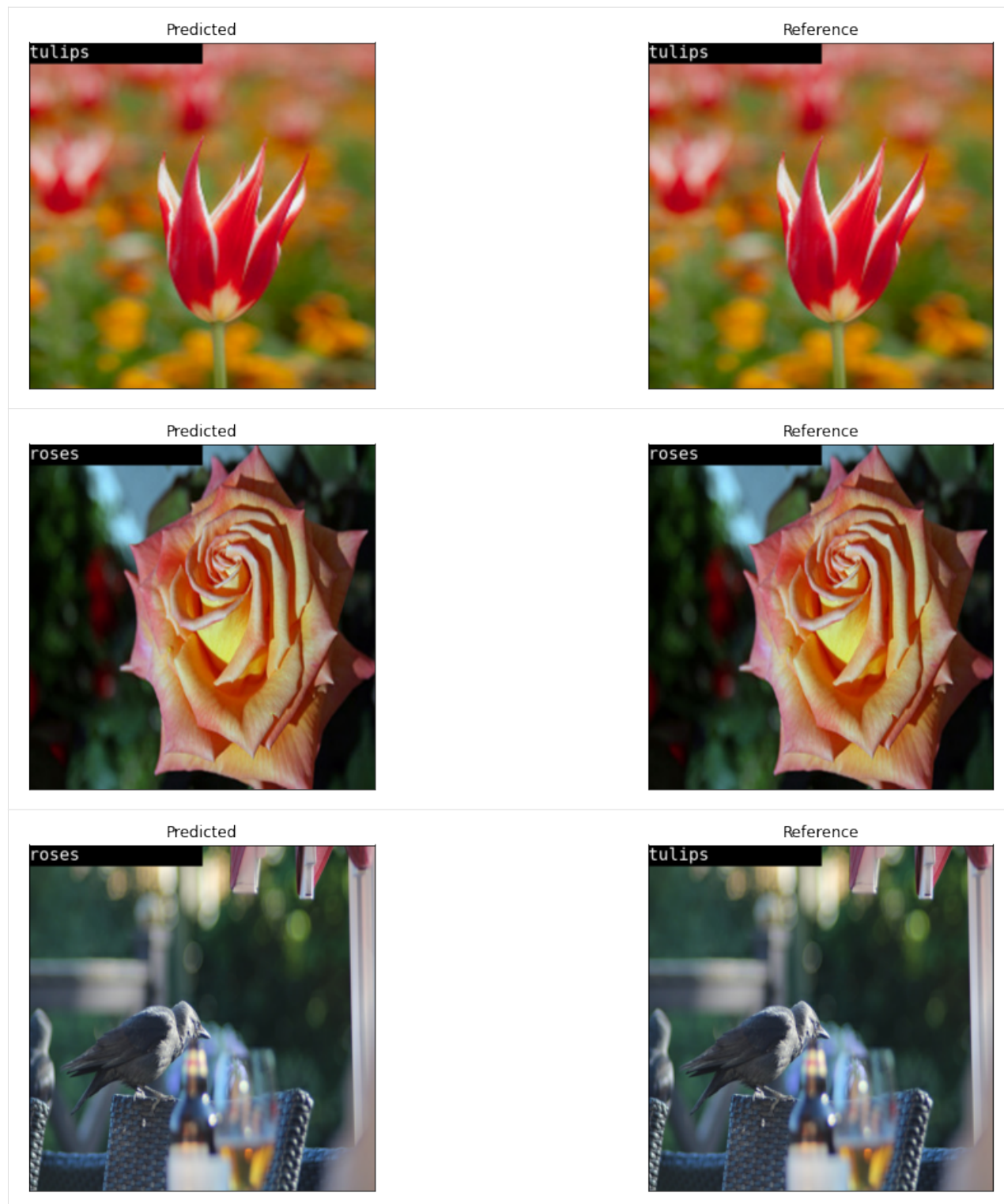












[ ]:

[ ]:

```
[ ]:
```

```
[ ]:
```

## 2.2 Example segmentation with DeepLabv3+

```
[1]: import numpy as np
      from glob import glob
      import tensorflow as tf
```

```
[2]: from tensorflow import keras
      from tensorflow.keras.layers import Input, Conv2D, Dropout, Activation, UpSampling2D,
      ↪GlobalMaxPooling2D, multiply
      from tensorflow.keras.backend import max
```

```
[3]: gpus = tf.config.experimental.list_physical_devices("GPU")
      #os.environ['TF_GPU_ALLOCATOR'] = 'cuda_malloc_async'

      if gpus:
          try:
              tf.config.experimental.set_virtual_device_configuration(
                  gpus[0],
                  [
                      tf.config.experimental.VirtualDeviceConfiguration(
                          memory_limit=1024 * 24
                      )
                  ],
              )
          except RuntimeError as e:
              print(e)
```

This example requires `swiss_army_keras` or higher:

```
pip3 install swiss_army_keras
```

```
[4]: #!pip3 install --upgrade git+https://github.com/waterviewsrl/swiss-army-keras.git
      from swiss_army_keras import models, base, utils, quantization_utils, training_utils,
      ↪dataset_utils
```

Matplotlib created a temporary config/cache directory at `/tmp/matplotlib-glcfk7fj` because the default path (`/.config/matplotlib`) is not a writable directory; it is highly recommended to set the `MPLCONFIGDIR` environment variable to a writable directory, in particular to speed up the import of Matplotlib and to better support multiprocessing.

```
[5]: # the indicator of a fresh run
      first_time_running = False
```

(continues on next page)

(continued from previous page)

```
# user-specified working directory
filepath = './oxford_iiit/'
```

## 2.2.1 The Oxford-IIIT Pets dataset

This example applies the dataset of [Oxford-IIIT Pet Dataset](#) (Parkhi et al. 2012). This dataset contains images of pets and their pixel-wise mask that indicates (1) pixels belonging to the pet, (2) pixels bordering the pet, and (3) surrounding pixels.

The Oxford-IIIT Pets dataset will be downloaded and unpacked through the cell blow.

```
[6]: if first_time_running:
    # downloading and executing data files
    import tarfile
    import urllib.request

    filename_image = filepath+'images.tar.gz'
    filename_target = filepath+'annotations.tar.gz'

    urllib.request.urlretrieve('http://www.robots.ox.ac.uk/~vgg/data/pets/data/images.
    ↪tar.gz', filename_image);
    urllib.request.urlretrieve('https://www.robots.ox.ac.uk/~vgg/data/pets/data/
    ↪annotations.tar.gz', filename_target);

    with tarfile.open(filename_image, "r:gz") as tar_io:
        tar_io.extractall(path=filepath)
    with tarfile.open(filename_target, "r:gz") as tar_io:
        tar_io.extractall(path=filepath)

    os.system(f'cd {filepath} && cd images && rm *.mat && cd ../../')
```

```
[7]: import albumentations as albu
```

```
[8]: from swiss_army_keras.dataset_utils import SegmentationAlbumentationsDataLoader
    from weathermentations.rain import DropsOnLens
```

```
[9]: n_labels = 3

input_tensor = tf.keras.layers.Input((256, 256, 3), batch_size=1)

X_decoder = models.deeplab_v3_plus(input_tensor,
                                   n_labels=n_labels,
                                   backbone='EfficientNetLiteB0',
                                   atrous_rates=[],
                                   deep_layer=5,
                                   activation='ReLU',
                                   batch_norm=True,
                                   pool=True,
```

(continues on next page)

(continued from previous page)

```

                                unpool=True,
                                name='Pets')

OUT_stack = X_decoder(input_tensor)
deeplabv3plus = keras.models.Model([input_tensor,], OUT_stack)
deeplabv3plus.preprocessing = X_decoder.preprocessing
deeplabv3plus.summary()

KerasTensor(type_spec=TensorSpec(shape=(1, 256, 256, 3), dtype=tf.float32, name='input_1
↪'), name='input_1', description="created by layer 'input_1'")
Model: "model_1"

```

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(1, 256, 256, 3)]	0
model (Functional)	(1, 256, 256, 3)	4652771

```

Total params: 4,652,771
Trainable params: 2,021,731
Non-trainable params: 2,631,040

```

[ ]:

```

[10]: def get_training_augmentation(width=513, height=513):
        train_transform = [

            #albu.RandomRotate90(p=0.5),

            #albu.OneOf([
            #
            #
            #], p=1),
            #albu.Resize(width, height, 2, p=1),

            #albu.ShiftScaleRotate(shift_limit = 0.05, scale_limit=(0.1, 0.1), ↪
↪interpolation=2, rotate_limit=25, p=0.3),

            albu.HorizontalFlip(p=0.5),
            #albu.VerticalFlip(p=0.15),

            albu.Compose([
            albu.OneOf([
                albu.ISONoise(p=1),
                albu.MultiplicativeNoise(p=1),
                albu.GaussNoise(p=1),
                albu.ImageCompression(p=1),

```

(continues on next page)

(continued from previous page)

```

        ],
        p=1,
    ),

    #albu.Perspective(pad_mode=2, p=0.1),

    albu.OneOf(
        [
            albu.CLAHE(p=1),
            albu.RandomBrightnessContrast(p=1),
            albu.Sharpen(p=1),
            albu.Emboss(p=1),
            albu.RandomGamma(p=1),
        ],
        p=0.5,
    ),

    albu.OneOf(
        [
            albu.Blur(blur_limit=5, p=1),
            albu.MedianBlur(blur_limit=5, p=1),
            albu.MotionBlur(blur_limit=5, p=1),
        ],
        p=0.25,
    ),

    albu.OneOf(
        [
            albu.RandomBrightnessContrast(p=1),
            albu.HueSaturationValue(p=1),
            albu.RGBShift(p=1),
        ],
        p=0.25,
    ),

    albu.OneOf(
        [
            #albu.ElasticTransform(alpha=10, sigma=10 * 0.05, alpha_affine=10 * 0.03,
→ p=0.5),
            albu.GridDistortion(p=0.5),
            albu.OpticalDistortion(distort_limit=0.3, shift_limit=0.05, p=1)
        ],
        p=0.25,
    ), p=0.5),

    #DropsOnLens(maxR=40, minR=30, p=0.1),
    albu.ToGray(p=0.2)
]

return albu.Compose(train_transform)
#return albu.Compose([])

```

(continues on next page)



(continued from previous page)

```

loader = SegmentationAlbumentationsDataLoader(num_classes=3,
                                              #precache=True,
                                              batch_size=16,
                                              width=256,
                                              height=256,
                                              dataset_path=filepath,
                                              masks_dir='annotations/trimaps',
                                              train_augmentations=get_training_
↪ augmentation(64, 64),
                                              label_shift=1,
                                              train_val_test_split=[0.8, 0.1, 0.1],
                                              normalization=deeplabv3plus.preprocessing,
                                              )

```

WARNING:root:Normalizing in range: [-1. 1.]

Train Images are good to go

```

WARNING:root:KEYS: None
WARNING:root:Router on "ipc:///tmp/swiss_army_keras_segmentation_clear_socket_HruiKtc="
WARNING:root:KEYS: None
WARNING:root:KEYS: (b'P+S690P{iVPfx<aFJwxFSY^ugFzjuW0naIh!o7J<', b':$80ST.hxA5xL7c+@
↪ $3YTEohOR^GhrJ2$zN@bR^')
WARNING:root:Dealer on "ipc:///tmp/swiss_army_keras_segmentation_clear_socket_HruiKtc="
WARNING:root:KEYS: None
WARNING:root:Attaching HOST CURVE keys.
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
↪ HruiKtc=): {'event': 128, 'value': 93, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↪ segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_CLOSED'}
WARNING:root:KEYS: None
WARNING:root:Dealer on "ipc:///tmp/swiss_army_keras_segmentation_clear_socket_HruiKtc="
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
↪ HruiKtc=): {'event': 4, 'value': 297, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↪ segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_CONNECT_RETRIED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
↪ HruiKtc=): {'event': 128, 'value': 94, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↪ segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_CLOSED'}
WARNING:root:Router on "ipc:///tmp/swiss_army_keras_segmentation_curve_socket_HruiKtc="
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
↪ HruiKtc=): {'event': 4, 'value': 297, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↪ segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_CONNECT_RETRIED'}
WARNING:root:Dealer on "ipc:///tmp/swiss_army_keras_segmentation_clear_socket_HruiKtc="
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
↪ HruiKtc=): {'event': 8, 'value': 104, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↪ segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_LISTENING'}
WARNING:root:KEYS: None
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_curve_socket_
↪ HruiKtc=): {'event': 8, 'value': 105, 'endpoint': b'ipc:///tmp/swiss_army_keras_
↪ segmentation_curve_socket_HruiKtc=', 'description': 'EVENT_LISTENING'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
↪ HruiKtc=): {'event': 1, 'value': 95, 'endpoint': b'ipc:///tmp/swiss_army_keras_

```

(continues on next page)

(continued from previous page)

```

→segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_CONNECTED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 32, 'value': 106, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_ACCEPTED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Dealer on "ipc:///tmp/swiss_army_keras_segmentation_clear_socket_HruiKtc="
WARNING:root:KEYS: None
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 1, 'value': 96, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_CONNECTED'}
WARNING:root:Dealer on "ipc:///tmp/swiss_army_keras_segmentation_clear_socket_HruiKtc="
WARNING:root:KEYS: None
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 32, 'value': 107, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_ACCEPTED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 1, 'value': 98, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_CONNECTED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 32, 'value': 108, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_ACCEPTED'}
WARNING:root:KEYS: None
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Dealer on "ipc:///tmp/swiss_army_keras_segmentation_clear_socket_HruiKtc="
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 1, 'value': 98, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_CONNECTED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 32, 'value': 111, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_ACCEPTED'}
WARNING:root:Dealer on "ipc:///tmp/swiss_army_keras_segmentation_clear_socket_HruiKtc="
WARNING:root:KEYS: None
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_

```

(continues on next page)



(continued from previous page)

```

→HruiKtc=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 1, 'value': 99, 'endpoint': b'ipc:///tmp/swiss_army_keras_
segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_CONNECTED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 32, 'value': 109, 'endpoint': b'ipc:///tmp/swiss_army_keras_
segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_ACCEPTED'}
WARNING:root:Dealer on "ipc:///tmp/swiss_army_keras_segmentation_clear_socket_HruiKtc="
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:KEYS: None
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 1, 'value': 93, 'endpoint': b'ipc:///tmp/swiss_army_keras_
segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_CONNECTED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 1, 'value': 101, 'endpoint': b'ipc:///tmp/swiss_army_keras_
segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_CONNECTED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 32, 'value': 122, 'endpoint': b'ipc:///tmp/swiss_army_keras_
segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_ACCEPTED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Dealer on "ipc:///tmp/swiss_army_keras_segmentation_clear_socket_HruiKtc="
WARNING:root:KEYS: None
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 1, 'value': 94, 'endpoint': b'ipc:///tmp/swiss_army_keras_
segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_CONNECTED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 32, 'value': 110, 'endpoint': b'ipc:///tmp/swiss_army_keras_
segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_ACCEPTED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 1, 'value': 101, 'endpoint': b'ipc:///tmp/swiss_army_keras_
segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_CONNECTED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_

```

(continues on next page)

(continued from previous page)

```

→HruiKtc=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 32, 'value': 124, 'endpoint': b'ipc:///tmp/swiss_army_keras_
segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_ACCEPTED'}
WARNING:root:Dealer on "ipc:///tmp/swiss_army_keras_segmentation_clear_socket_HruiKtc="
WARNING:root:KEYS: None
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 1, 'value': 102, 'endpoint': b'ipc:///tmp/swiss_army_keras_
segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_CONNECTED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 32, 'value': 112, 'endpoint': b'ipc:///tmp/swiss_army_keras_
segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_ACCEPTED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Dealer on "ipc:///tmp/swiss_army_keras_segmentation_clear_socket_HruiKtc="
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 32, 'value': 113, 'endpoint': b'ipc:///tmp/swiss_army_keras_
segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_ACCEPTED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 1, 'value': 103, 'endpoint': b'ipc:///tmp/swiss_army_keras_
segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_CONNECTED'}
WARNING:root:KEYS: None
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 32, 'value': 127, 'endpoint': b'ipc:///tmp/swiss_army_keras_
segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_ACCEPTED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Dealer on "ipc:///tmp/swiss_army_keras_segmentation_clear_socket_HruiKtc="
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 1, 'value': 104, 'endpoint': b'ipc:///tmp/swiss_army_keras_
segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_CONNECTED'}
WARNING:root:KEYS: None
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 32, 'value': 114, 'endpoint': b'ipc:///tmp/swiss_army_keras_
segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_ACCEPTED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_

```

(continues on next page)

(continued from previous page)

```

→segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Dealer on "ipc:///tmp/swiss_army_keras_segmentation_clear_socket_HruiKtc="
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 32, 'value': 115, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_ACCEPTED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 1, 'value': 106, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_CONNECTED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:KEYS: None
WARNING:root:Dealer on "ipc:///tmp/swiss_army_keras_segmentation_clear_socket_HruiKtc="
WARNING:root:KEYS: None
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 32, 'value': 116, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_ACCEPTED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 1, 'value': 106, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_CONNECTED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Dealer on "ipc:///tmp/swiss_army_keras_segmentation_clear_socket_HruiKtc="
WARNING:root:KEYS: None
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 32, 'value': 117, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_ACCEPTED'}
WARNING:root:KEYS: None
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Dealer on "ipc:///tmp/swiss_army_keras_segmentation_clear_socket_HruiKtc="
WARNING:root:Dealer on "ipc:///tmp/swiss_army_keras_segmentation_clear_socket_HruiKtc="
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 1, 'value': 107, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_CONNECTED'}
WARNING:root:KEYS: None
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 1, 'value': 111, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_CONNECTED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_

```

(continues on next page)

(continued from previous page)

```

→segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 32, 'value': 118, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_ACCEPTED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 1, 'value': 108, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_CONNECTED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Dealer on "ipc:///tmp/swiss_army_keras_segmentation_clear_socket_HruiKtc="
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 32, 'value': 119, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_ACCEPTED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 1, 'value': 109, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_CONNECTED'}
WARNING:root:KEYS: None
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 32, 'value': 134, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_ACCEPTED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Dealer on "ipc:///tmp/swiss_army_keras_segmentation_clear_socket_HruiKtc="
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 32, 'value': 120, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_ACCEPTED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 1, 'value': 110, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_CONNECTED'}
WARNING:root:Event Monitor(Host: ipc:///tmp/swiss_army_keras_segmentation_clear_socket_
→HruiKtc=): {'event': 4096, 'value': 0, 'endpoint': b'ipc:///tmp/swiss_army_keras_
→segmentation_clear_socket_HruiKtc=', 'description': 'EVENT_HANDSHAKE_SUCCEEDED'}

```

```
[11]: train_set, val_set, test_set = loader.build_datasets()
```



```
[12]: loader.show_images(dset='train', num_images=16)
```

Image



Mask



Image



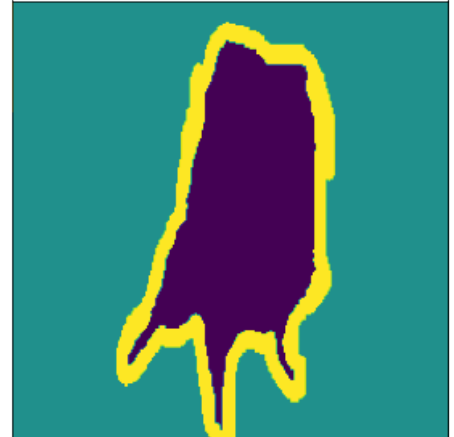
Mask

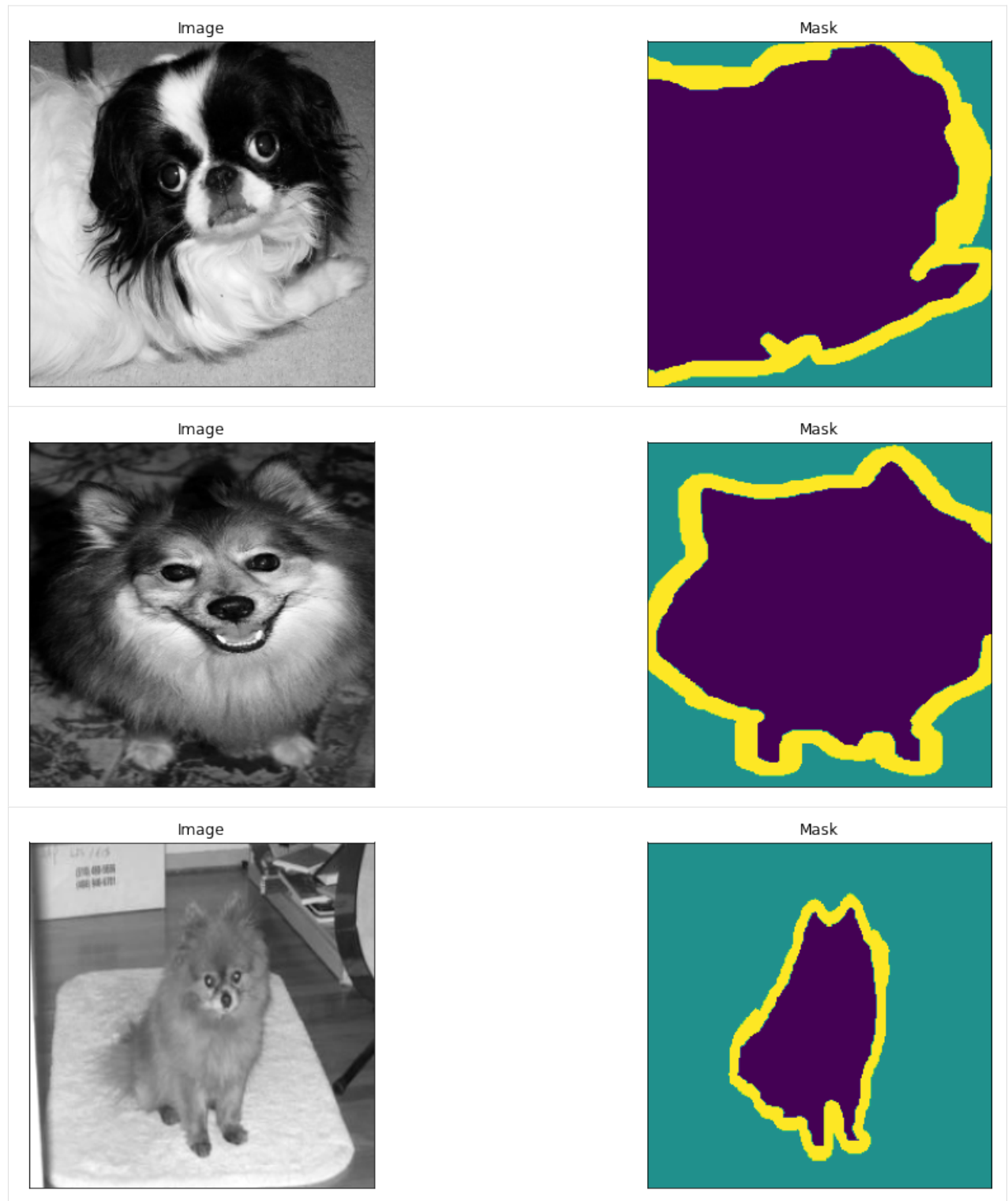


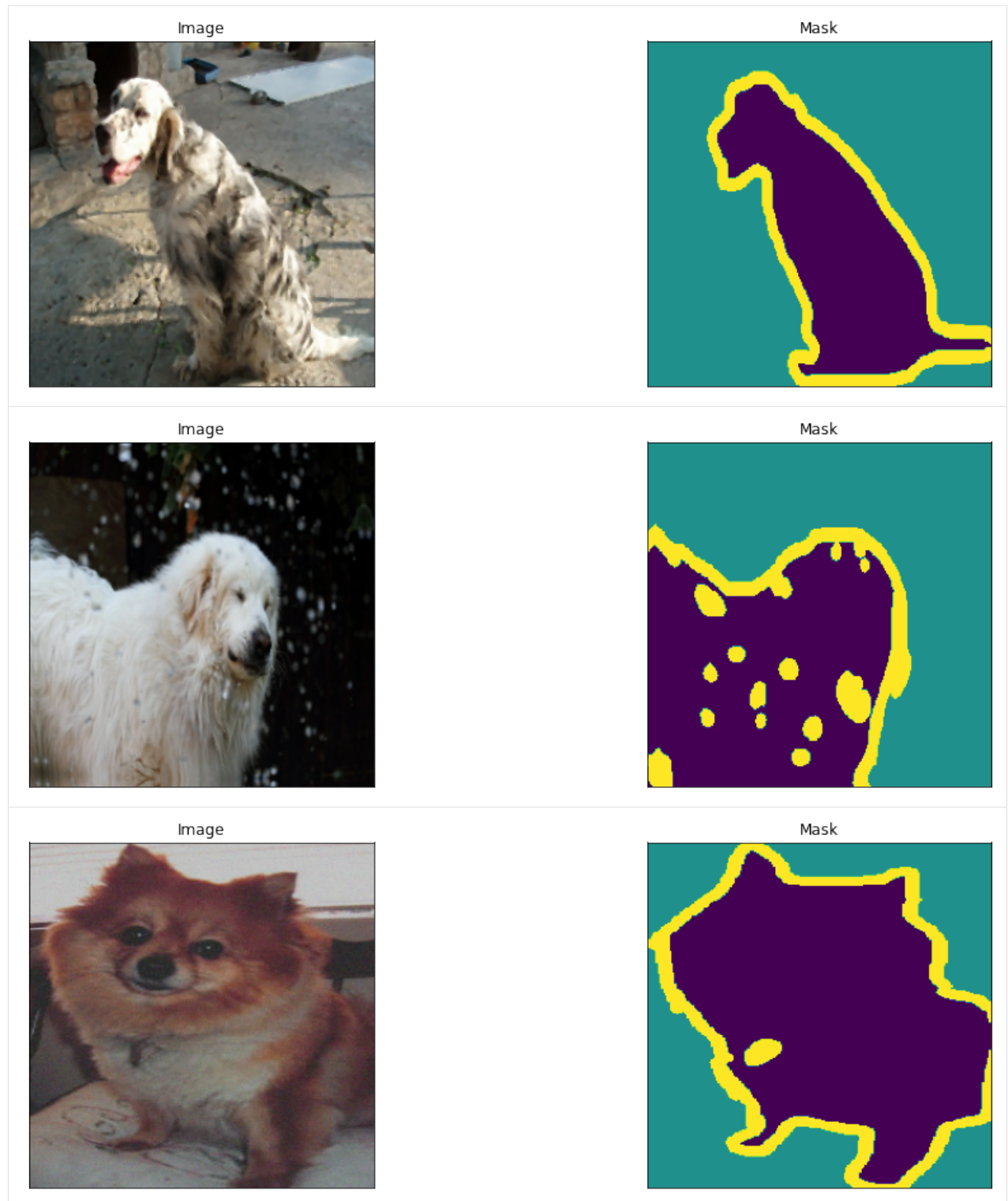
Image



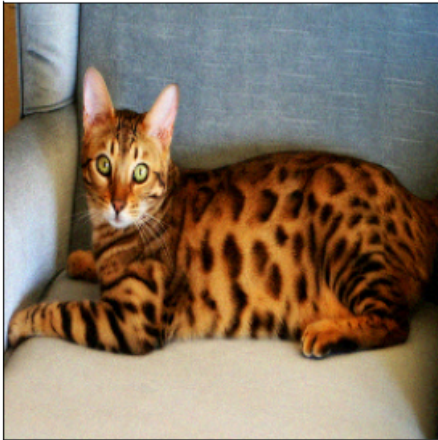
Mask







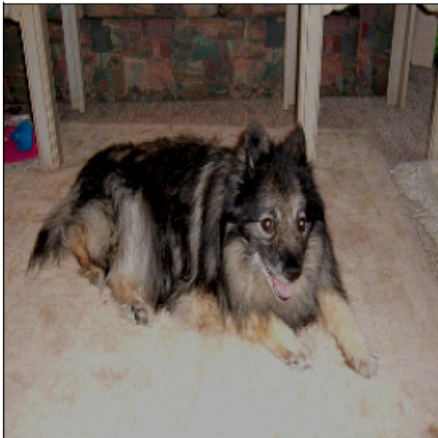
Image



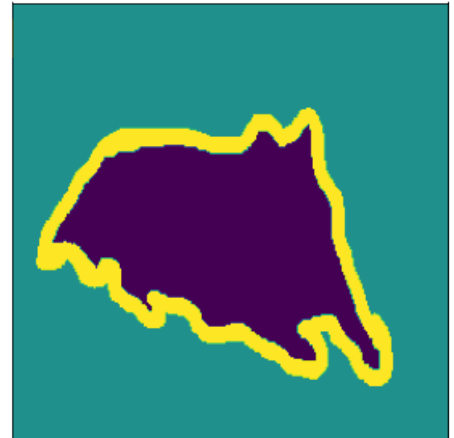
Mask



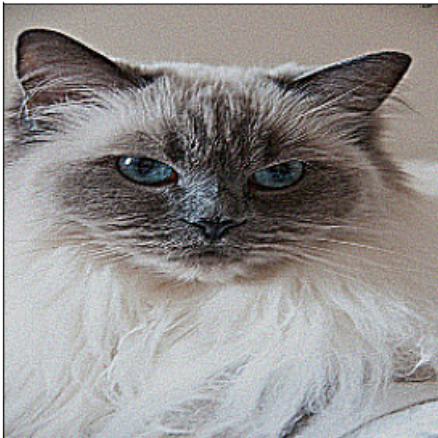
Image



Mask



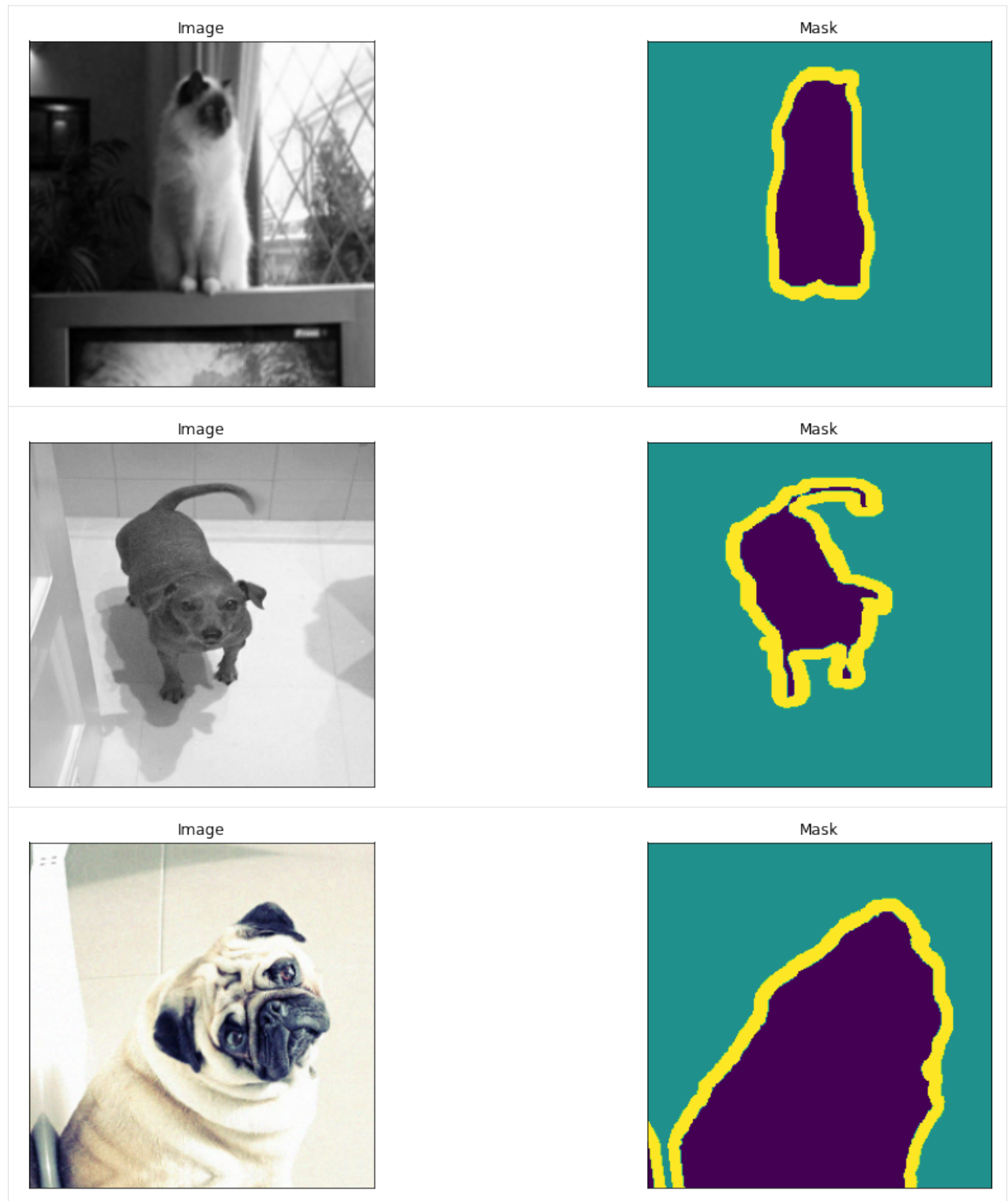
Image

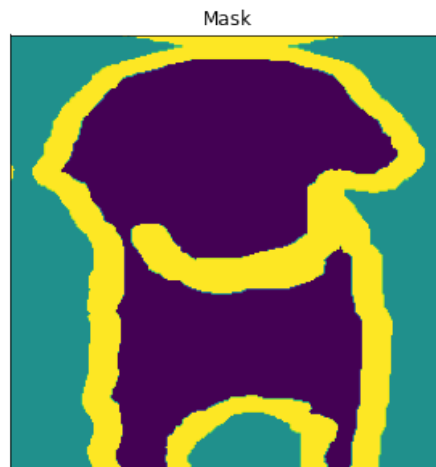
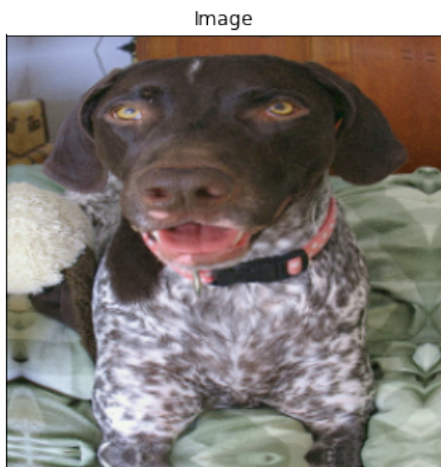


Mask









```
[13]: from swiss_army_keras import losses, metrics
```

```
[14]: loss = losses.loss_adapter(losses.symmetrical_unified_focal_loss(), from_logits=True,
    ↪ label_smoothing=0.1)
    accuracy = tf.keras.metrics.CategoricalAccuracy()
    miou = metrics.MeanIOU(num_classes=n_labels)
```

```
[15]: from swiss_army_keras.training_utils import TrainingDriver
    from adabelief_tf import AdaBeliefOptimizer
```

```
[16]: td = TrainingDriver(deeplabv3plus, 'pets_model', AdaBeliefOptimizer(), loss, [accuracy,
    ↪ miou], train_set, val_set, test_set, 20 )
```

Please check your arguments if you have upgraded adabelief-tf from version 0.0.1.

Modifications to default arguments:

	eps	weight_decouple	rectify
adabelief-tf=0.0.1	1e-08	Not supported	Not supported
>=0.1.0 (Current 0.2.1)	1e-14	supported	default: True
SGD better than Adam (e.g. CNN for Image Classification)		Adam better than SGD (e.g. ↪ Transformer, GAN)	

↪ Recommended epsilon = 1e-7 Recommended epsilon = 1e-14  
 For a complete table of recommended hyperparameters, see  
<https://github.com/juntang-zhuang/Adabelief-Optimizer>  
 You can disable the log message by setting "print\_change\_log = False", though it is ↪  
 ↪ recommended to keep as a reminder.

```
[17]: td.run()
```

Model: "model\_1"

Layer (type)	Output Shape	Param #
=====		

(continues on next page)

(continued from previous page)

```

input_1 (InputLayer)      [(1, 256, 256, 3)]      0
=====
model (Functional)        (1, 256, 256, 3)        4652771
=====
Total params: 4,652,771
Trainable params: 2,021,731
Non-trainable params: 2,631,040
=====
Epoch 1/20
WARNING:tensorflow:Model was constructed with shape (1, 256, 256, 3) for input_
↳KerasTensor(type_spec=TensorSpec(shape=(1, 256, 256, 3), dtype=tf.float32, name='input_
↳1'), name='input_1', description="created by layer 'input_1'"), but it was called on_
↳an input with incompatible shape (16, 256, 256, 3).

/usr/local/lib/python3.6/dist-packages/keras/utils/generic_utils.py:497:
↳CustomMaskWarning: Custom mask layers require a config and must override get_config.
↳When loading, the custom mask layer must be passed to the custom_objects argument.
category=CustomMaskWarning)
WARNING:tensorflow:Model was constructed with shape (1, 256, 256, 3) for input_
↳KerasTensor(type_spec=TensorSpec(shape=(1, 256, 256, 3), dtype=tf.float32, name='input_
↳1'), name='input_1', description="created by layer 'input_1'"), but it was called on_
↳an input with incompatible shape (16, 256, 256, 3).

WARNING:tensorflow:Model was constructed with shape (1, 256, 256, 3) for input_
↳KerasTensor(type_spec=TensorSpec(shape=(1, 256, 256, 3), dtype=tf.float32, name='input_
↳1'), name='input_1', description="created by layer 'input_1'"), but it was called on_
↳an input with incompatible shape (16, 256, 256, 3).

WARNING:tensorflow:Model was constructed with shape (1, 256, 256, 3) for input_
↳KerasTensor(type_spec=TensorSpec(shape=(1, 256, 256, 3), dtype=tf.float32, name='input_
↳1'), name='input_1', description="created by layer 'input_1'"), but it was called on_
↳an input with incompatible shape (16, 256, 256, 3).

WARNING:tensorflow:Model was constructed with shape (1, 256, 256, 3) for input_
↳KerasTensor(type_spec=TensorSpec(shape=(1, 256, 256, 3), dtype=tf.float32, name='input_
↳1'), name='input_1', description="created by layer 'input_1'"), but it was called on_
↳an input with incompatible shape (16, 256, 256, 3).

WARNING:tensorflow:Model was constructed with shape (1, 256, 256, 3) for input_
↳KerasTensor(type_spec=TensorSpec(shape=(1, 256, 256, 3), dtype=tf.float32, name='input_
↳1'), name='input_1', description="created by layer 'input_1'"), but it was called on_
↳an input with incompatible shape (16, 256, 256, 3).

WARNING:tensorflow:Model was constructed with shape (1, 256, 256, 3) for input_
↳KerasTensor(type_spec=TensorSpec(shape=(1, 256, 256, 3), dtype=tf.float32, name='input_
↳1'), name='input_1', description="created by layer 'input_1'"), but it was called on_
↳an input with incompatible shape (16, 256, 256, 3).

WARNING:tensorflow:Model was constructed with shape (1, 256, 256, 3) for input_
↳KerasTensor(type_spec=TensorSpec(shape=(1, 256, 256, 3), dtype=tf.float32, name='input_
↳1'), name='input_1', description="created by layer 'input_1'"), but it was called on_
↳an input with incompatible shape (16, 256, 256, 3).

WARNING:tensorflow:Model was constructed with shape (1, 256, 256, 3) for input_
↳KerasTensor(type_spec=TensorSpec(shape=(1, 256, 256, 3), dtype=tf.float32, name='input_
↳1'), name='input_1', description="created by layer 'input_1'"), but it was called on_
↳an input with incompatible shape (16, 256, 256, 3).
    
```

(continues on next page)

(continued from previous page)

```

↳an input with incompatible shape (16, 256, 256, 3).
WARNING:tensorflow:Model was constructed with shape (1, 256, 256, 3) for input_
↳KerasTensor(type_spec=TensorSpec(shape=(1, 256, 256, 3), dtype=tf.float32, name='input_
↳1'), name='input_1', description="created by layer 'input_1'"), but it was called on_
↳an input with incompatible shape (16, 256, 256, 3).
WARNING:tensorflow:Model was constructed with shape (1, 256, 256, 3) for input_
↳KerasTensor(type_spec=TensorSpec(shape=(1, 256, 256, 3), dtype=tf.float32, name='input_
↳1'), name='input_1', description="created by layer 'input_1'"), but it was called on_
↳an input with incompatible shape (16, 256, 256, 3).
WARNING:tensorflow:Model was constructed with shape (1, 256, 256, 3) for input_
↳KerasTensor(type_spec=TensorSpec(shape=(1, 256, 256, 3), dtype=tf.float32, name='input_
↳1'), name='input_1', description="created by layer 'input_1'"), but it was called on_
↳an input with incompatible shape (16, 256, 256, 3).
369/369 [=====] - ETA: 0s - loss: 0.4020 - categorical_accuracy:
↳0.8556 - mean_iou: 0.6432WARNING:tensorflow:Model was constructed with shape (1, 256,
↳256, 3) for input KerasTensor(type_spec=TensorSpec(shape=(1, 256, 256, 3), dtype=tf.
↳float32, name='input_1'), name='input_1', description="created by layer 'input_1'"),
↳but it was called on an input with incompatible shape (16, 256, 256, 3).
WARNING:tensorflow:Model was constructed with shape (1, 256, 256, 3) for input_
↳KerasTensor(type_spec=TensorSpec(shape=(1, 256, 256, 3), dtype=tf.float32, name='input_
↳1'), name='input_1', description="created by layer 'input_1'"), but it was called on_
↳an input with incompatible shape (16, 256, 256, 3).
WARNING:tensorflow:Model was constructed with shape (1, 256, 256, 3) for input_
↳KerasTensor(type_spec=TensorSpec(shape=(1, 256, 256, 3), dtype=tf.float32, name='input_
↳1'), name='input_1', description="created by layer 'input_1'"), but it was called on_
↳an input with incompatible shape (16, 256, 256, 3).
WARNING:tensorflow:Model was constructed with shape (1, 256, 256, 3) for input_
↳KerasTensor(type_spec=TensorSpec(shape=(1, 256, 256, 3), dtype=tf.float32, name='input_
↳1'), name='input_1', description="created by layer 'input_1'"), but it was called on_
↳an input with incompatible shape (16, 256, 256, 3).
WARNING:tensorflow:Model was constructed with shape (1, 256, 256, 3) for input_
↳KerasTensor(type_spec=TensorSpec(shape=(1, 256, 256, 3), dtype=tf.float32, name='input_
↳1'), name='input_1', description="created by layer 'input_1'"), but it was called on_
↳an input with incompatible shape (16, 256, 256, 3).
WARNING:tensorflow:Model was constructed with shape (1, 256, 256, 3) for input_
↳KerasTensor(type_spec=TensorSpec(shape=(1, 256, 256, 3), dtype=tf.float32, name='input_
↳1'), name='input_1', description="created by layer 'input_1'"), but it was called on_
↳an input with incompatible shape (16, 256, 256, 3).
369/369 [=====] - 34s 72ms/step - loss: 0.4020 - categorical_
↳accuracy: 0.8556 - mean_iou: 0.6432 - val_loss: 0.3942 - val_categorical_accuracy: 0.
↳8722 - val_mean_iou: 0.6910
Epoch 2/20
369/369 [=====] - 25s 68ms/step - loss: 0.3573 - categorical_
↳accuracy: 0.8992 - mean_iou: 0.7252 - val_loss: 0.3513 - val_categorical_accuracy: 0.
↳9059 - val_mean_iou: 0.7555
Epoch 3/20
369/369 [=====] - 25s 68ms/step - loss: 0.3472 - categorical_

```

(continues on next page)

(continued from previous page)

```

↪accuracy: 0.9085 - mean_iou: 0.7450 - val_loss: 0.3609 - val_categorical_accuracy: 0.
↪8975 - val_mean_iou: 0.7307

```

Epoch 4/20

```

369/369 [=====] - 25s 68ms/step - loss: 0.3427 - categorical_
↪accuracy: 0.9128 - mean_iou: 0.7541 - val_loss: 0.3413 - val_categorical_accuracy: 0.
↪9147 - val_mean_iou: 0.7666

```

Epoch 5/20

```

369/369 [=====] - 25s 68ms/step - loss: 0.3392 - categorical_
↪accuracy: 0.9159 - mean_iou: 0.7611 - val_loss: 0.3381 - val_categorical_accuracy: 0.
↪9161 - val_mean_iou: 0.7653

```

Epoch 6/20

```

369/369 [=====] - 25s 68ms/step - loss: 0.3365 - categorical_
↪accuracy: 0.9185 - mean_iou: 0.7671 - val_loss: 0.3359 - val_categorical_accuracy: 0.
↪9190 - val_mean_iou: 0.7677

```

Epoch 7/20

```

369/369 [=====] - 25s 68ms/step - loss: 0.3338 - categorical_
↪accuracy: 0.9206 - mean_iou: 0.7719 - val_loss: 0.3448 - val_categorical_accuracy: 0.
↪9122 - val_mean_iou: 0.7718

```

Epoch 8/20

```

369/369 [=====] - 25s 68ms/step - loss: 0.3324 - categorical_
↪accuracy: 0.9220 - mean_iou: 0.7749 - val_loss: 0.3363 - val_categorical_accuracy: 0.
↪9185 - val_mean_iou: 0.7703

```

Epoch 9/20

```

369/369 [=====] - 25s 69ms/step - loss: 0.3311 - categorical_
↪accuracy: 0.9231 - mean_iou: 0.7774 - val_loss: 0.3293 - val_categorical_accuracy: 0.
↪9246 - val_mean_iou: 0.7814

```

Epoch 10/20

```

369/369 [=====] - 26s 70ms/step - loss: 0.3290 - categorical_
↪accuracy: 0.9247 - mean_iou: 0.7811 - val_loss: 0.3298 - val_categorical_accuracy: 0.
↪9246 - val_mean_iou: 0.7860

```

Epoch 11/20

```

369/369 [=====] - 26s 70ms/step - loss: 0.3277 - categorical_
↪accuracy: 0.9260 - mean_iou: 0.7841 - val_loss: 0.3350 - val_categorical_accuracy: 0.
↪9196 - val_mean_iou: 0.7772

```

Epoch 12/20

```

369/369 [=====] - 26s 70ms/step - loss: 0.3270 - categorical_
↪accuracy: 0.9264 - mean_iou: 0.7854 - val_loss: 0.3396 - val_categorical_accuracy: 0.
↪9155 - val_mean_iou: 0.7709

```

Epoch 13/20

```

369/369 [=====] - 26s 70ms/step - loss: 0.3249 - categorical_
↪accuracy: 0.9283 - mean_iou: 0.7895 - val_loss: 0.3311 - val_categorical_accuracy: 0.
↪9233 - val_mean_iou: 0.7847

```

Epoch 14/20

```

369/369 [=====] - 26s 70ms/step - loss: 0.3248 - categorical_
↪accuracy: 0.9283 - mean_iou: 0.7897 - val_loss: 0.3342 - val_categorical_accuracy: 0.
↪9208 - val_mean_iou: 0.7768

```

Epoch 15/20

```

369/369 [=====] - 26s 70ms/step - loss: 0.3232 - categorical_
↪accuracy: 0.9296 - mean_iou: 0.7924 - val_loss: 0.3303 - val_categorical_accuracy: 0.
↪9235 - val_mean_iou: 0.7871

```

Epoch 16/20

```

369/369 [=====] - 26s 70ms/step - loss: 0.3226 - categorical_

```

(continues on next page)

(continued from previous page)

```

→accuracy: 0.9302 - mean_iou: 0.7943 - val_loss: 0.3273 - val_categorical_accuracy: 0.
→9269 - val_mean_iou: 0.7877

```

Epoch 17/20

```

369/369 [=====] - 26s 70ms/step - loss: 0.3213 - categorical_
→accuracy: 0.9313 - mean_iou: 0.7963 - val_loss: 0.3287 - val_categorical_accuracy: 0.
→9251 - val_mean_iou: 0.7878

```

Epoch 18/20

```

369/369 [=====] - 26s 70ms/step - loss: 0.3207 - categorical_
→accuracy: 0.9319 - mean_iou: 0.7979 - val_loss: 0.3296 - val_categorical_accuracy: 0.
→9248 - val_mean_iou: 0.7900

```

Epoch 19/20

```

369/369 [=====] - 26s 70ms/step - loss: 0.3192 - categorical_
→accuracy: 0.9331 - mean_iou: 0.8008 - val_loss: 0.3265 - val_categorical_accuracy: 0.
→9270 - val_mean_iou: 0.7960

```

Epoch 20/20

```

369/369 [=====] - 26s 70ms/step - loss: 0.3182 - categorical_
→accuracy: 0.9339 - mean_iou: 0.8023 - val_loss: 0.3279 - val_categorical_accuracy: 0.
→9259 - val_mean_iou: 0.7865

```

WARNING:root:Unfreezing Model

Model: "model\_1"

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(1, 256, 256, 3)]	0
model (Functional)	(1, 256, 256, 3)	4652771

Total params: 4,652,771

Trainable params: 4,611,299

Non-trainable params: 41,472

Epoch 1/20

```

WARNING:tensorflow:Model was constructed with shape (1, 256, 256, 3) for input_
→KerasTensor(type_spec=TensorSpec(shape=(1, 256, 256, 3), dtype=tf.float32, name='input_
→1'), name='input_1', description="created by layer 'input_1'"), but it was called on_
→an input with incompatible shape (16, 256, 256, 3).

```

```

WARNING:tensorflow:Model was constructed with shape (1, 256, 256, 3) for input_
→KerasTensor(type_spec=TensorSpec(shape=(1, 256, 256, 3), dtype=tf.float32, name='input_
→1'), name='input_1', description="created by layer 'input_1'"), but it was called on_
→an input with incompatible shape (16, 256, 256, 3).

```

```

WARNING:tensorflow:Model was constructed with shape (1, 256, 256, 3) for input_
→KerasTensor(type_spec=TensorSpec(shape=(1, 256, 256, 3), dtype=tf.float32, name='input_
→1'), name='input_1', description="created by layer 'input_1'"), but it was called on_
→an input with incompatible shape (16, 256, 256, 3).

```

```

WARNING:tensorflow:Model was constructed with shape (1, 256, 256, 3) for input_
→KerasTensor(type_spec=TensorSpec(shape=(1, 256, 256, 3), dtype=tf.float32, name='input_
→1'), name='input_1', description="created by layer 'input_1'"), but it was called on_
→an input with incompatible shape (16, 256, 256, 3).

```



```
WARNING:tensorflow:Model was constructed with shape (1, 256, 256, 3) for input_
↳KerasTensor(type_spec=TensorSpec(shape=(1, 256, 256, 3), dtype=tf.float32, name='input_
↳1'), name='input_1', description="created by layer 'input_1'"), but it was called on_
↳an input with incompatible shape (16, 256, 256, 3).
```

```
WARNING:tensorflow:Model was constructed with shape (1, 256, 256, 3) for input_
↳KerasTensor(type_spec=TensorSpec(shape=(1, 256, 256, 3), dtype=tf.float32, name='input_
↳1'), name='input_1', description="created by layer 'input_1'"), but it was called on_
↳an input with incompatible shape (16, 256, 256, 3).
```

```
WARNING:tensorflow:Model was constructed with shape (1, 256, 256, 3) for input_
↳KerasTensor(type_spec=TensorSpec(shape=(1, 256, 256, 3), dtype=tf.float32, name='input_
↳1'), name='input_1', description="created by layer 'input_1'"), but it was called on_
↳an input with incompatible shape (16, 256, 256, 3).
```

```
WARNING:tensorflow:Model was constructed with shape (1, 256, 256, 3) for input_
↳KerasTensor(type_spec=TensorSpec(shape=(1, 256, 256, 3), dtype=tf.float32, name='input_
↳1'), name='input_1', description="created by layer 'input_1'"), but it was called on_
↳an input with incompatible shape (16, 256, 256, 3).
```

```
WARNING:tensorflow:Model was constructed with shape (1, 256, 256, 3) for input_
↳KerasTensor(type_spec=TensorSpec(shape=(1, 256, 256, 3), dtype=tf.float32, name='input_
↳1'), name='input_1', description="created by layer 'input_1'"), but it was called on_
↳an input with incompatible shape (16, 256, 256, 3).
```

```
WARNING:tensorflow:Model was constructed with shape (1, 256, 256, 3) for input_
↳KerasTensor(type_spec=TensorSpec(shape=(1, 256, 256, 3), dtype=tf.float32, name='input_
↳1'), name='input_1', description="created by layer 'input_1'"), but it was called on_
↳an input with incompatible shape (16, 256, 256, 3).
```

```
WARNING:tensorflow:Model was constructed with shape (1, 256, 256, 3) for input_
↳KerasTensor(type_spec=TensorSpec(shape=(1, 256, 256, 3), dtype=tf.float32, name='input_
↳1'), name='input_1', description="created by layer 'input_1'"), but it was called on_
↳an input with incompatible shape (16, 256, 256, 3).
```

```
WARNING:tensorflow:Model was constructed with shape (1, 256, 256, 3) for input_
↳KerasTensor(type_spec=TensorSpec(shape=(1, 256, 256, 3), dtype=tf.float32, name='input_
↳1'), name='input_1', description="created by layer 'input_1'"), but it was called on_
↳an input with incompatible shape (16, 256, 256, 3).
```

```
369/369 [=====] - ETA: 0s - loss: 0.3725 - categorical_accuracy:
↳0.8885 - mean_iou: 0.7141WARNING:tensorflow:Model was constructed with shape (1, 256,
↳256, 3) for input KerasTensor(type_spec=TensorSpec(shape=(1, 256, 256, 3), dtype=tf.
↳float32, name='input_1'), name='input_1', description="created by layer 'input_1'"),
↳but it was called on an input with incompatible shape (16, 256, 256, 3).
```

```
WARNING:tensorflow:Model was constructed with shape (1, 256, 256, 3) for input_
↳KerasTensor(type_spec=TensorSpec(shape=(1, 256, 256, 3), dtype=tf.float32, name='input_
↳1'), name='input_1', description="created by layer 'input_1'"), but it was called on_
↳an input with incompatible shape (16, 256, 256, 3).
```

```
WARNING:tensorflow:Model was constructed with shape (1, 256, 256, 3) for input_
↳KerasTensor(type_spec=TensorSpec(shape=(1, 256, 256, 3), dtype=tf.float32, name='input_
↳1'), name='input_1', description="created by layer 'input_1'"), but it was called on_
↳an input with incompatible shape (16, 256, 256, 3).
```

```
WARNING:tensorflow:Model was constructed with shape (1, 256, 256, 3) for input_
↳KerasTensor(type_spec=TensorSpec(shape=(1, 256, 256, 3), dtype=tf.float32, name='input_
```

(continues on next page)



(continued from previous page)

```
→1'), name='input_1', description="created by layer 'input_1'"), but it was called on_
→an input with incompatible shape (16, 256, 256, 3).
```

```
WARNING:tensorflow:Model was constructed with shape (1, 256, 256, 3) for input_
→KerasTensor(type_spec=TensorSpec(shape=(1, 256, 256, 3), dtype=tf.float32, name='input_
→1'), name='input_1', description="created by layer 'input_1'"), but it was called on_
→an input with incompatible shape (16, 256, 256, 3).
```

```
WARNING:tensorflow:Model was constructed with shape (1, 256, 256, 3) for input_
→KerasTensor(type_spec=TensorSpec(shape=(1, 256, 256, 3), dtype=tf.float32, name='input_
→1'), name='input_1', description="created by layer 'input_1'"), but it was called on_
→an input with incompatible shape (16, 256, 256, 3).
```

```
369/369 [=====] - 79s 158ms/step - loss: 0.3725 - categorical_
→accuracy: 0.8885 - mean_iou: 0.7141 - val_loss: 0.3536 - val_categorical_accuracy: 0.
→9022 - val_mean_iou: 0.7398
```

Epoch 2/20

```
369/369 [=====] - 54s 147ms/step - loss: 0.3466 - categorical_
→accuracy: 0.9086 - mean_iou: 0.7514 - val_loss: 0.3413 - val_categorical_accuracy: 0.
→9133 - val_mean_iou: 0.7657
```

Epoch 3/20

```
369/369 [=====] - 54s 147ms/step - loss: 0.3382 - categorical_
→accuracy: 0.9162 - mean_iou: 0.7665 - val_loss: 0.3346 - val_categorical_accuracy: 0.
→9198 - val_mean_iou: 0.7775
```

Epoch 4/20

```
369/369 [=====] - 55s 148ms/step - loss: 0.3327 - categorical_
→accuracy: 0.9211 - mean_iou: 0.7763 - val_loss: 0.3299 - val_categorical_accuracy: 0.
→9233 - val_mean_iou: 0.7848
```

Epoch 5/20

```
369/369 [=====] - 54s 147ms/step - loss: 0.3298 - categorical_
→accuracy: 0.9237 - mean_iou: 0.7820 - val_loss: 0.3296 - val_categorical_accuracy: 0.
→9239 - val_mean_iou: 0.7882
```

Epoch 6/20

```
369/369 [=====] - 54s 147ms/step - loss: 0.3274 - categorical_
→accuracy: 0.9256 - mean_iou: 0.7861 - val_loss: 0.3270 - val_categorical_accuracy: 0.
→9259 - val_mean_iou: 0.7867
```

Epoch 7/20

```
369/369 [=====] - 54s 147ms/step - loss: 0.3244 - categorical_
→accuracy: 0.9285 - mean_iou: 0.7923 - val_loss: 0.3286 - val_categorical_accuracy: 0.
→9250 - val_mean_iou: 0.7894
```

Epoch 8/20

```
369/369 [=====] - 54s 146ms/step - loss: 0.3231 - categorical_
→accuracy: 0.9295 - mean_iou: 0.7942 - val_loss: 0.3310 - val_categorical_accuracy: 0.
→9223 - val_mean_iou: 0.7835
```

Epoch 9/20

```
369/369 [=====] - 55s 148ms/step - loss: 0.3209 - categorical_
→accuracy: 0.9314 - mean_iou: 0.7983 - val_loss: 0.3263 - val_categorical_accuracy: 0.
→9270 - val_mean_iou: 0.7944
```

Epoch 10/20

```
369/369 [=====] - 54s 145ms/step - loss: 0.3197 - categorical_
→accuracy: 0.9324 - mean_iou: 0.8006 - val_loss: 0.3268 - val_categorical_accuracy: 0.
→9261 - val_mean_iou: 0.7907
```

Epoch 11/20

(continues on next page)

(continued from previous page)

```
369/369 [=====] - 54s 147ms/step - loss: 0.3182 - categorical_
→accuracy: 0.9337 - mean_iou: 0.8034 - val_loss: 0.3222 - val_categorical_accuracy: 0.
→9304 - val_mean_iou: 0.7989
```

Epoch 12/20

```
369/369 [=====] - 54s 147ms/step - loss: 0.3183 - categorical_
→accuracy: 0.9338 - mean_iou: 0.8039 - val_loss: 0.3264 - val_categorical_accuracy: 0.
→9273 - val_mean_iou: 0.7977
```

Epoch 13/20

```
369/369 [=====] - 54s 148ms/step - loss: 0.3168 - categorical_
→accuracy: 0.9350 - mean_iou: 0.8064 - val_loss: 0.3264 - val_categorical_accuracy: 0.
→9276 - val_mean_iou: 0.7952
```

Epoch 14/20

```
369/369 [=====] - 54s 147ms/step - loss: 0.3149 - categorical_
→accuracy: 0.9365 - mean_iou: 0.8101 - val_loss: 0.3242 - val_categorical_accuracy: 0.
→9292 - val_mean_iou: 0.7985
```

Epoch 15/20

```
369/369 [=====] - 55s 148ms/step - loss: 0.3142 - categorical_
→accuracy: 0.9371 - mean_iou: 0.8115 - val_loss: 0.3274 - val_categorical_accuracy: 0.
→9272 - val_mean_iou: 0.7959
```

Epoch 16/20

```
369/369 [=====] - 54s 147ms/step - loss: 0.3145 - categorical_
→accuracy: 0.9367 - mean_iou: 0.8109 - val_loss: 0.3215 - val_categorical_accuracy: 0.
→9309 - val_mean_iou: 0.8016
```

Epoch 17/20

```
369/369 [=====] - 54s 147ms/step - loss: 0.3130 - categorical_
→accuracy: 0.9380 - mean_iou: 0.8135 - val_loss: 0.3233 - val_categorical_accuracy: 0.
→9295 - val_mean_iou: 0.7984
```

Epoch 18/20

```
369/369 [=====] - 54s 147ms/step - loss: 0.3117 - categorical_
→accuracy: 0.9391 - mean_iou: 0.8162 - val_loss: 0.3201 - val_categorical_accuracy: 0.
→9322 - val_mean_iou: 0.8038
```

Epoch 19/20

```
369/369 [=====] - 54s 147ms/step - loss: 0.3136 - categorical_
→accuracy: 0.9375 - mean_iou: 0.8128 - val_loss: 0.3252 - val_categorical_accuracy: 0.
→9280 - val_mean_iou: 0.7962
```

Epoch 20/20

```
369/369 [=====] - 54s 147ms/step - loss: 0.3119 - categorical_
→accuracy: 0.9389 - mean_iou: 0.8158 - val_loss: 0.3201 - val_categorical_accuracy: 0.
→9322 - val_mean_iou: 0.8035
```

WARNING:root:Quantizing Model

WARNING:tensorflow:Compiled the loaded model, but the compiled metrics have yet to be built. `model.compile\_metrics` will be empty until you train or evaluate the model.

WARNING:tensorflow:Compiled the loaded model, but the compiled metrics have yet to be built. `model.compile\_metrics` will be empty until you train or evaluate the model.

INFO:tensorflow:Assets written to: /tmp/tmpuq2\_6gro/assets

INFO:tensorflow:Assets written to: /tmp/tmpuq2\_6gro/assets

/usr/local/lib/python3.6/dist-packages/keras/utils/generic\_utils.py:497: CustomMaskWarning: Custom mask layers require a config and must override get\_config. When loading, the custom mask layer must be passed to the custom\_objects argument.

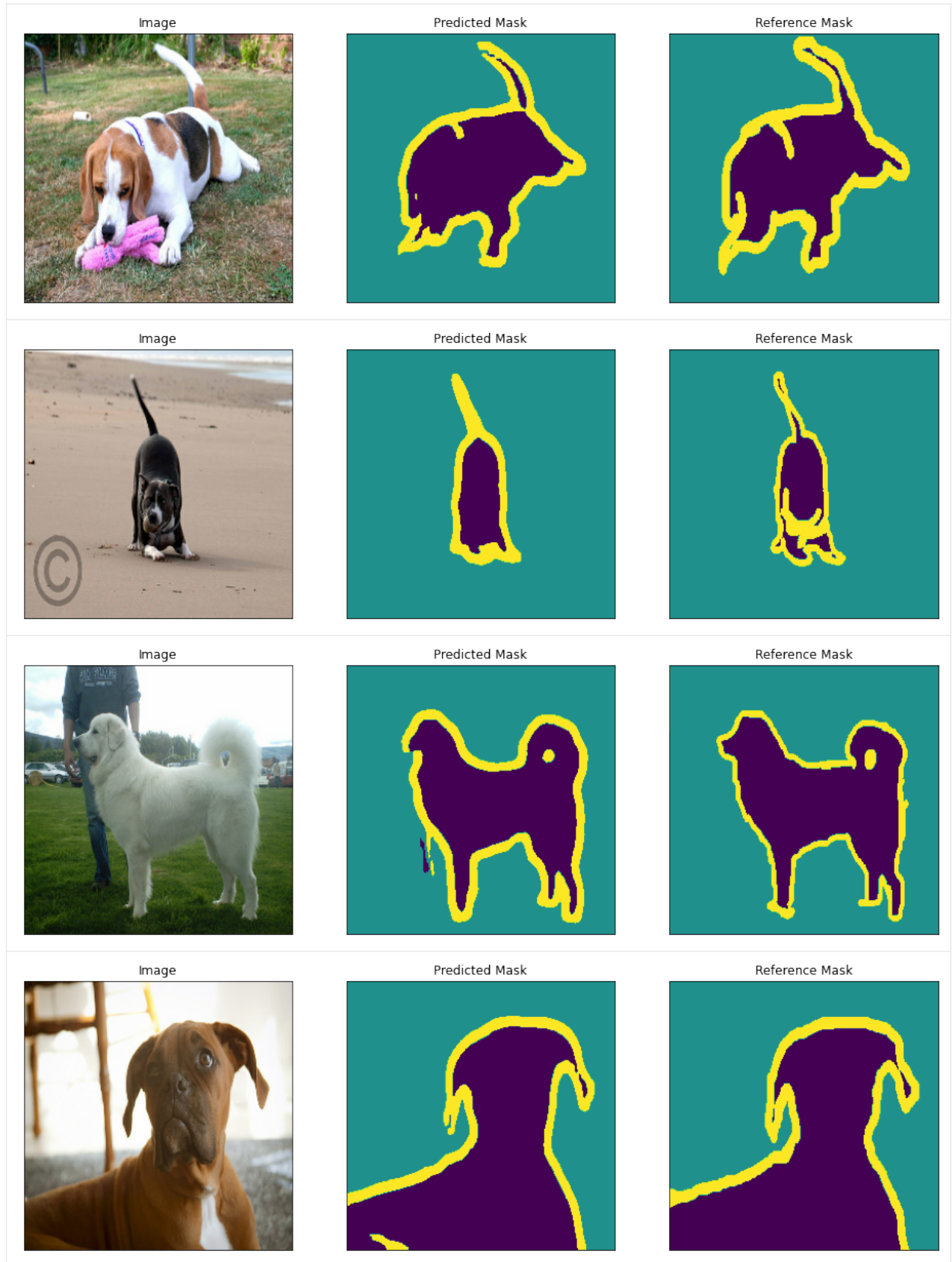
(continues on next page)

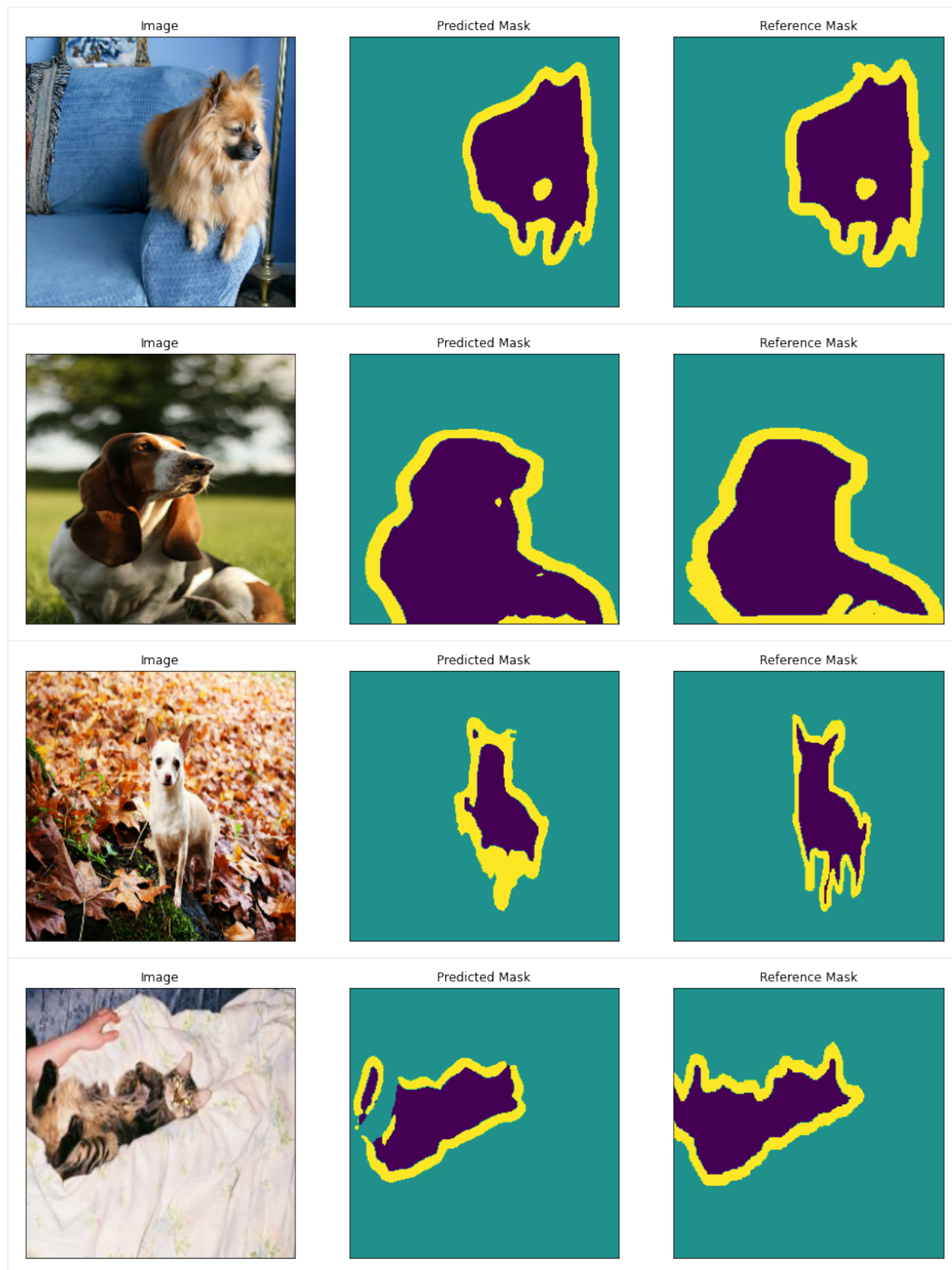
(continued from previous page)

```
category=CustomMaskWarning)
WARNING:absl:For model inputs containing unsupported operations which cannot be
↳quantized, the `inference_input_type` attribute will default to the original type.
WARNING:tensorflow:Compiled the loaded model, but the compiled metrics have yet to be
↳built. `model.compile_metrics` will be empty until you train or evaluate the model.
WARNING:tensorflow:Compiled the loaded model, but the compiled metrics have yet to be
↳built. `model.compile_metrics` will be empty until you train or evaluate the model.
INFO:tensorflow:Assets written to: /tmp/tmpobro38yf/assets
INFO:tensorflow:Assets written to: /tmp/tmpobro38yf/assets
/usr/local/lib/python3.6/dist-packages/keras/utils/generic_utils.py:497:
↳CustomMaskWarning: Custom mask layers require a config and must override get_config.
↳When loading, the custom mask layer must be passed to the custom_objects argument.
category=CustomMaskWarning)
<Figure size 432x288 with 0 Axes>
```

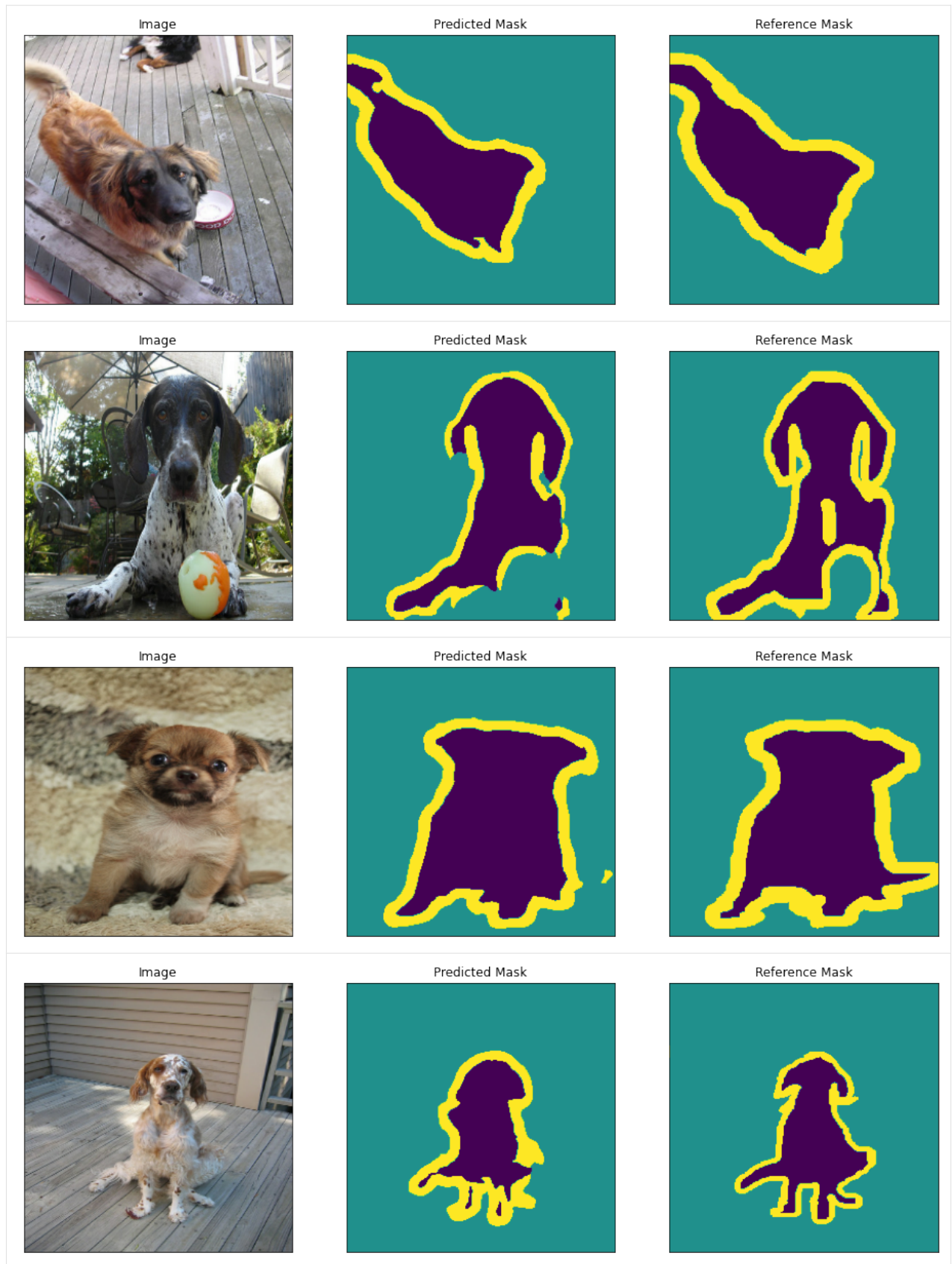
```
[18]: loader.show_results(deeplabv3plus, 16)
```

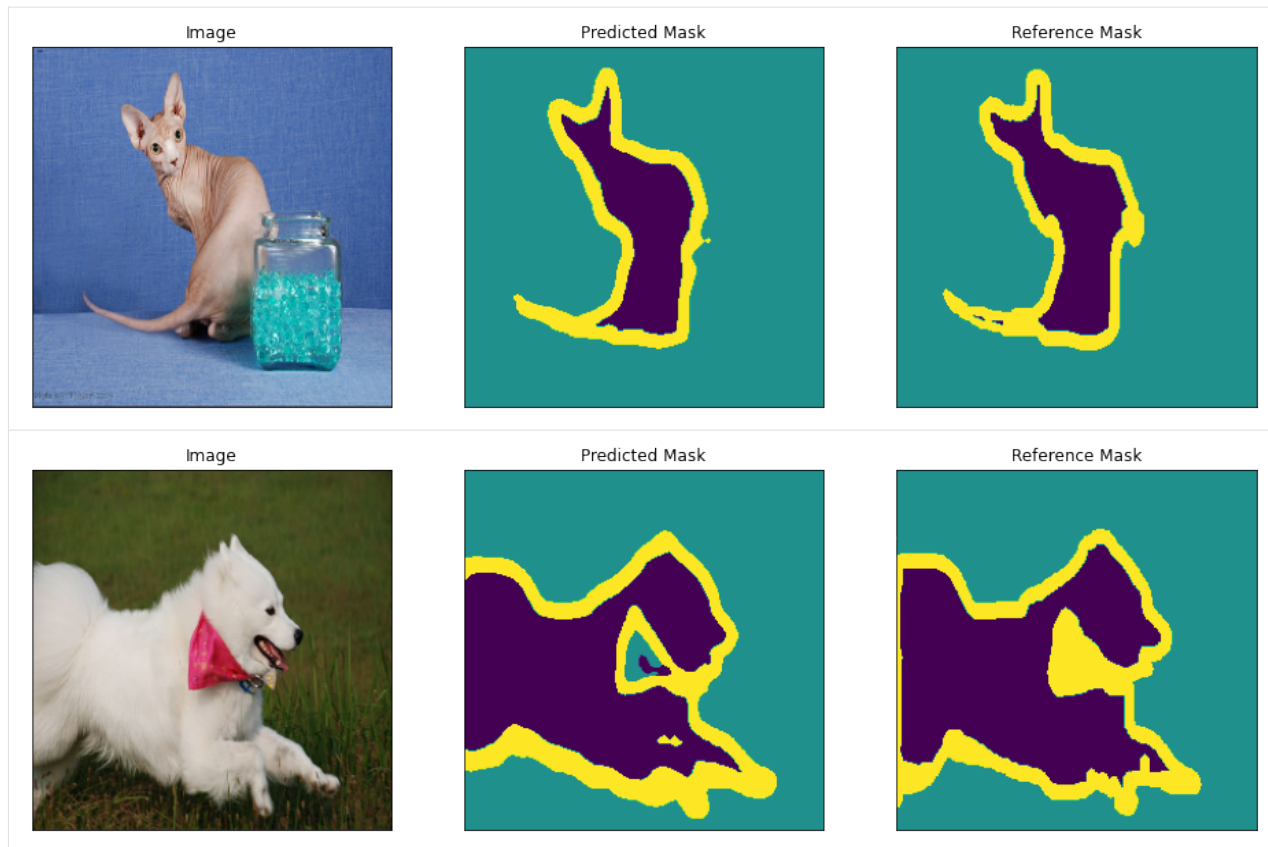












```
[ ]:
```

## 2.3 Model User Guide

```
[ ]: This user guide requires `keras-unet-collection==0.1.9` or higher.
```

```
## Content

* [**U-net**] (#U-net)
* [**V-net**] (#V-net)
* [**Attention-Unet**] (#Attention-Unet)
* [**U-net++**] (#U-net++)
* [**UNET 3+**] (#UNET-3+)
* [**R2U-net**] (#R2U-net)
* [**ResUnet-a**] (#ResUnet-a)
* [**U^2-Net**] (#U^2-Net)
* [**TransUNET**] (#TransUNET)
* [**Swin-UNET**] (#Swin-UNET)
```

```
[1]: import tensorflow as tf
      from tensorflow import keras
```



```
[2]: print('TensorFlow {}; Keras {}'.format(tf.__version__, keras.__version__))
TensorFlow 2.5.0; Keras 2.5.0
```

### 2.3.1 Step 1: importing models from swiss\_army\_keras

```
[3]: from swiss_army_keras import models
```

### 2.3.2 Step 2: defining your hyper-parameters

Commonly used hyper-parameter options are listed as follows. Full details are available through the Python helper function:

- **inpu<sub>st</sub>\_size**: a tuple or list that defines the shape of input tensors.
  - `models.resnet_a_2d`, `models.transnet_2d`, and `models.swin_unet_2d` support int only, others also support `inpust_size=(None, None, 3)`.
  - `activation='PReLU'` is not compatible with `inpust_size=(None, None, 3)`.
- **filter\_num**: a list that defines the number of convolutional filters per down- and up-sampling blocks.
  - For `unet_2d`, `att_unet_2d`, `unet_plus_2d`, `r2_unet_2d`, `depth ≥ 2` is expected.
  - For `resnet_a_2d` and `u2net_2d`, `depth ≥ 3` is expected.
- **n\_labels**: number of output targets, e.g., `n_labels=2` for binary classification.
- **activation**: the activation function of hidden layers. Available choices are 'ReLU', 'LeakyReLU', 'PReLU', 'ELU', 'GELU', 'Snake'.
- **output\_activation**: the activation function of the output layer. Recommended choices are 'Sigmoid', 'Softmax', None (linear), 'Snake'.
- **batch\_norm**: if specified as True, all convolutional layers will be configured as stacks of “Conv2D-BN-Activation”.
- **stack\_num\_down**: number of convolutional layers per downsampling level.
- **stack\_num\_up**: number of convolutional layers (after concatenation) per upsampling level.
- **pool**: the configuration of downsampling (encoding) blocks.
  - `pool=False`: downsampling with a convolutional layer (2-by-2 convolution kernels with 2 strides; optional batch normalization and activation).
  - `pool=True` or `pool='max'` downsampling with a max-pooling layer.
  - `pool='ave'` downsampling with a average-pooling layer.
- **unpool**: the configuration of upsampling (decoding) blocks.
  - `unpool=False`: upsampling with a transpose convolutional layer (2-by-2 convolution kernels with 2 strides; optional batch normalization and activation).
  - `unpool=True` or `unpool='bilinear'` upsampling with bilinear interpolation.
  - `unpool='nearest'` upsampling with reflective padding.
- **name**: user-specified prefix of the configured layer and model. Use `keras.models.Model.summary` to identify the exact name of each layer.

### 2.3.3 Step 3: Configuring your model

#### Note

Configured models can be saved through `model.save(filepath, save_traces=True)`, but they may contain python objects that are not part of the `tensorflow.keras`. Thus when loading the model, it is preferred to load the weights only, and set/freeze them within a new configuration.

e.g.

```
weights = dummy_loader(model_old_path)
model_new = swin_transformer_model(...)
model_new.set_weights(weights)
```

#### U-net

**Example 1:** U-net for binary classification with:

1. Five down- and upsampling levels (or four downsampling levels and one bottom level).
2. Two convolutional layers per downsampling level.
3. One convolutional layer (after concatenation) per upsampling level.
4. Gaussian Error Linear Unit (GELU) activation, Softmax output activation, batch normalization.
5. Downsampling through Maxpooling.
6. Upsampling through reflective padding.

```
[4]: model = models.unet_2d((None, None, 3), [64, 128, 256, 512, 1024], n_labels=2,
                             stack_num_down=2, stack_num_up=1,
                             activation='GELU', output_activation='Softmax',
                             batch_norm=True, pool='max', unpool='nearest', name='unet')
```

#### V-net

**Example 2:** Vnet (originally proposed for 3-d inputs, here modified for 2-d inputs) for binary classification with:

1. Input size of (256, 256, 1); PReLU does not support input tensor with shapes of NoneType
2. Five down- and upsampling levels (or four downsampling levels and one bottom level).
3. Number of stacked convolutional layers of the residual path increase with downsampling levels from one to three (symmetrically, decrease with upsampling levels).
  - `res_num_ini=1`
  - `res_num_max=3`
4. PReLU activation, Softmax output activation, batch normalization.
5. Downsampling through stride convolutional layers.
6. Upsampling through transpose convolutional layers.

```
[5]: model = models.vnet_2d((256, 256, 1), filter_num=[16, 32, 64, 128, 256], n_labels=2,
                             res_num_ini=1, res_num_max=3,
                             activation='PReLU', output_activation='Softmax',
                             batch_norm=True, pool=False, unpool=False, name='vnet')
```

## Attention-Unet

**Example 3:** attention-Unet for single target regression with:

1. Four down- and upsampling levels.
2. Two convolutional layers per downsampling level.
3. Two convolutional layers (after concatenation) per upsampling level.
4. ReLU activation, linear output activation (None), batch normalization.
5. Additive attention, ReLU attention activation.
6. Downsampling through stride convolutional layers.
7. Upsampling through bilinear interpolation.

```
[6]: model = models.att_unet_2d((None, None, 3), [64, 128, 256, 512], n_labels=1,
                                stack_num_down=2, stack_num_up=2,
                                activation='ReLU', atten_activation='ReLU', attention='add',
                                ↪output_activation=None,
                                batch_norm=True, pool=False, unpool='bilinear', name='attunet',
                                ↪')
```

## U-net++

**Example 4:** U-net++ for three-label classification with:

1. Four down- and upsampling levels.
2. Two convolutional layers per downsampling level.
3. Two convolutional layers (after concatenation) per upsampling level.
4. LeakyReLU activation, Softmax output activation, no batch normalization.
5. Downsampling through Maxpooling.
6. Upsampling through transpose convolutional layers.
7. Deep supervision.

```
[7]: model = models.unet_plus_2d((None, None, 3), [64, 128, 256, 512], n_labels=3,
                                stack_num_down=2, stack_num_up=2,
                                activation='LeakyReLU', output_activation='Softmax',
                                batch_norm=False, pool='max', unpool=False, deep_
                                ↪supervision=True, name='xnet')
```

```
-----
deep_supervision = True
names of output tensors are listed as follows ("sup0" is the shallowest supervision_
↪layer;
```

(continues on next page)

(continued from previous page)

"final" is the final output layer):

```
xnet_output_sup0_activation
xnet_output_sup1_activation
xnet_output_sup2_activation
xnet_output_final_activation
```

## UNET 3+

**Example 5:** UNet 3+ for binary classification with:

1. Four down- and upsampling levels.
2. Two convolutional layers per downsampling level.
3. One convolutional layers (after concatenation) per upsampling level.
4. ReLU activation, Sigmoid output activation, batch normalization.
5. Downsampling through Maxpooling.
6. Upsampling through transpose convolutional layers.
7. Deep supervision.

```
[8]: model = models.unet_3plus_2d((128, 128, 3), n_labels=2, filter_num_down=[64, 128, 256, ↵
↵512],
                                filter_num_skip='auto', filter_num_aggregate='auto',
                                stack_num_down=2, stack_num_up=1, activation='ReLU', output_
↵activation='Sigmoid',
                                batch_norm=True, pool='max', unpool=False, deep_
↵supervision=True, name='unet3plus')
```

Automated hyper-parameter determination is applied with the following details:

```
-----
    Number of convolution filters after each full-scale skip connection: filter_num_
↵skip = [64, 64, 64]
    Number of channels of full-scale aggregated feature maps: filter_num_aggregate = ↵
↵256
-----
```

deep\_supervision = True

names of output tensors are listed as follows ("sup0" is the shallowest supervision\_
↵layer;

"final" is the final output layer):

```
unet3plus_output_sup0_activation
unet3plus_output_sup1_activation
unet3plus_output_sup2_activation
unet3plus_output_final_activation
```

- filter\_num\_skip and filter\_num\_aggregate can be specified explicitly:

```
[9]: model = models.unet_3plus_2d((128, 128, 3), n_labels=2, filter_num_down=[64, 128, 256, ↵
↵512],
                                filter_num_skip=[64, 64, 64], filter_num_aggregate=256,
```

(continues on next page)

(continued from previous page)

```

stack_num_down=2, stack_num_up=1, activation='ReLU', output_
activation='Sigmoid',
batch_norm=True, pool='max', unpool=False, deep_
supervision=True, name='unet3plus')

-----
deep_supervision = True
names of output tensors are listed as follows ("sup0" is the shallowest supervision_
layer;
"final" is the final output layer):

    unet3plus_output_sup0_activation
    unet3plus_output_sup1_activation
    unet3plus_output_sup2_activation
    unet3plus_output_final_activation
    
```

## R2U-net

**Example 6:** R2U-net for binary classification with:

1. Four down- and upsampling levels.
2. Two recurrent convolutional layers with two iterations per down- and upsampling level.
3. ReLU activation, Softmax output activation, no batch normalization.
4. Downsampling through Maxpooling.
5. Upsampling through reflective padding.

```

[10]: model = models.r2_unet_2d((None, None, 3), [64, 128, 256, 512], n_labels=2,
                                stack_num_down=2, stack_num_up=1, recur_num=2,
                                activation='ReLU', output_activation='Softmax',
                                batch_norm=True, pool='max', unpool='nearest', name='r2unet')
    
```

## ResUnet-a

**Example 7:** ResUnet-a for 16-label classification with:

1. input size of (128, 128, 3)
2. Six downsampling levels followed by an Atrous Spatial Pyramid Pooling (ASPP) layer with 256 filters.
3. Six upsampling levels followed by an ASPP layer with 128 filters.
4. dilation rates of {1, 3, 15, 31} for shallow layers, {1,3,15} for intermediate layers, and {1,} for deep layers.
5. ReLU activation, Sigmoid output activation, batch normalization.
6. Downsampling through stride convolutional layers.
7. Upsampling through reflective padding.

```

[11]: model = models.resunet_a_2d((128, 128, 3), [32, 64, 128, 256, 512, 1024],
                                   dilation_num=[1, 3, 15, 31],
                                   n_labels=16, aspp_num_down=256, aspp_num_up=128,
    
```

(continues on next page)

(continued from previous page)

```

activation='ReLU', output_activation='Sigmoid',
batch_norm=True, pool=False, unpool='nearest', name='resunet
↪')

```

Received dilation rates: [1, 3, 15, 31]

Received dilation rates are not defined on a per downsampling level basis.

Automated determinations are applied with the following details:

```

depth-0, dilation_rate = [1, 3, 15, 31]
depth-1, dilation_rate = [1, 3, 15, 31]
depth-2, dilation_rate = [1, 3, 15]
depth-3, dilation_rate = [1, 3, 15]
depth-4, dilation_rate = [1]
depth-5, dilation_rate = [1]

```

- dilation\_num can be specified per down- and upampling level:

```

[12]: model = models.resunet_a2d((128, 128, 3), [32, 64, 128, 256, 512, 1024],
                                dilation_num=[[1, 3, 15, 31], [1, 3, 15, 31], [1, 3, 15], [1,
↪ 3, 15], [1,], [1,],],
                                n_labels=16, aspp_num_down=256, aspp_num_up=128,
                                activation='ReLU', output_activation='Sigmoid',
                                batch_norm=True, pool=False, unpool='nearest', name='resunet
↪')

```

## U^2-Net

**Example 8:** U^2-Net for binary classification with:

1. Six downsampling levels with the first four layers built with RSU, and the last two (one downsampling layer, one bottom layer) built with RSU-F4.
  - filter\_num\_down=[64, 128, 256, 512]
  - filter\_mid\_num\_down=[32, 32, 64, 128]
  - filter\_4f\_num=[512, 512]
  - filter\_4f\_mid\_num=[256, 256]
2. Six upsampling levels with the deepest layer built with RSU-F4, and the other four layers built with RSU.
  - filter\_num\_up=[64, 64, 128, 256]
  - filter\_mid\_num\_up=[16, 32, 64, 128]
3. ReLU activation, Sigmoid output activation, batch normalization.
4. Deep supervision
5. Downsampling through stride convolutional layers.
6. Upsampling through transpose convolutional layers.

\*In the original work of U^2-Net, down- and upsampling were achieved through maxpooling (pool=True or pool='max') and bilinear interpolation (unpool=True or unpool='bilinear').



```
[13]: model = models.u2net_2d((128, 128, 3), n_labels=2,
                             filter_num_down=[64, 128, 256, 512], filter_num_up=[64, 64, 128,
                             ↪ 256],
                             filter_mid_num_down=[32, 32, 64, 128], filter_mid_num_up=[16, 32,
                             ↪ 64, 128],
                             filter_4f_num=[512, 512], filter_4f_mid_num=[256, 256],
                             activation='ReLU', output_activation=None,
                             batch_norm=True, pool=False, unpool=False, deep_supervision=True,
                             ↪ name='u2net')
```

```
-----
The depth of u2net_2d = len(filter_num_down) + len(filter_4f_num) = 6
-----
```

deep\_supervision = True

names of output tensors are listed as follows ("sup0" is the shallowest supervision,  
 ↪ layer;

"final" is the final output layer):

```
u2net_output_sup0_trans_conv
u2net_output_sup1_trans_conv
u2net_output_sup2_trans_conv
u2net_output_sup3_trans_conv
u2net_output_sup4_trans_conv
u2net_output_sup5_trans_conv
u2net_output_final
```

- u2net\_2d supports automated determination of filter numbers per down- and upsampling level. Auto-mode may produce a slightly larger network.

```
[14]: model = models.u2net_2d((None, None, 3), n_labels=2,
                             filter_num_down=[64, 128, 256, 512],
                             activation='ReLU', output_activation='Sigmoid',
                             batch_norm=True, pool=False, unpool=False, deep_supervision=True,
                             ↪ name='u2net')
```

Automated hyper-parameter determination is applied with the following details:

```
-----
Number of RSU output channels within downsampling blocks: filter_num_down = [64,
↪ 128, 256, 512]
Number of RSU intermediate channels within downsampling blocks: filter_mid_num_
↪ down = [16, 32, 64, 128]
Number of RSU output channels within upsampling blocks: filter_num_up = [64, 128,
↪ 256, 512]
Number of RSU intermediate channels within upsampling blocks: filter_mid_num_up_
↪ = [16, 32, 64, 128]
Number of RSU-4F output channels within downsampling and bottom blocks: filter_
↪ 4f_num = [512, 512]
Number of RSU-4F intermediate channels within downsampling and bottom blocks:
↪ filter_4f_num = [256, 256]
-----
```

Explicitly specifying keywords listed above if their "auto" settings do not satisfy your,  
 ↪ needs

```
-----
The depth of u2net_2d = len(filter_num_down) + len(filter_4f_num) = 6
```

(continues on next page)

(continued from previous page)

```

-----
deep_supervision = True
names of output tensors are listed as follows ("sup0" is the shallowest supervision_
↳ layer;
"final" is the final output layer):

    u2net_output_sup0_activation
    u2net_output_sup1_activation
    u2net_output_sup2_activation
    u2net_output_sup3_activation
    u2net_output_sup4_activation
    u2net_output_sup5_activation
    u2net_output_final_activation

```

## TransUNET

**Example 9:** TransUNET for 12-label classification with:

- input size of (512, 512, 3)
- Four down- and upsampling levels.
- Two convolutional layers per downsampling level.
- Two convolutional layers (after concatenation) per upsampling level.
- 12 transformer blocks (num\_transformer=12).
- 12 attention heads (num\_heads=12).
- 3072 MLP nodes per vision transformer (num\_mlp=3072).
- 768 embedding dimensions (embed\_dim=768).
- Gaussian Error Linear Unit (GELU) activation for transformer MLPs.
- ReLU activation, softmax output activation, batch normalization.
- Downsampling through maxpooling.
- Upsampling through bilinear interpolation.

```

[15]: model = models.transunet_2d((512, 512, 3), filter_num=[64, 128, 256, 512], n_labels=12,
↳ stack_num_down=2, stack_num_up=2,
    embed_dim=768, num_mlp=3072, num_heads=12, num_
↳ transformer=12,
    activation='ReLU', mlp_activation='GELU', output_
↳ activation='Softmax',
    batch_norm=True, pool=True, unpool='bilinear', name=
↳ 'transunet')

```

## Swin-UNET

**Example 10:** Swin-UNET for 3-label classification with:

- input size of (128, 128, 3)
- Four down- and upsampling levels (or three downsampling levels and one bottom level) (depth=4).
- Two Swin-Transformers per downsampling level.
- Two Swin-Transformers (after concatenation) per upsampling level.
- Extract 2-by-2 patches from the input (patch\_size=(2, 2))
- Embed 2-by-2 patches to 64 dimensions (filter\_num\_begin=64, a.k.a, number of embedded dimensions).
- Number of attention heads for each down- and upsampling level: num\_heads=[4, 8, 8, 8].
- Size of attention windows for each down- and upsampling level: window\_size=[4, 2, 2, 2].
- 512 nodes per Swin-Transformer (num\_mlp=512)
- Shift attention windows (i.e., Swin-MSA) (shift\_window=True).

```
[16]: model = models.swin_unet_2d((128, 128, 3), filter_num_begin=64, n_labels=3, depth=4,
↳ stack_num_down=2, stack_num_up=2,
patch_size=(2, 2), num_heads=[4, 8, 8, 8], window_size=[4, 2,
↳ 2, 2], num_mlp=512,
output_activation='Softmax', shift_window=True, name='swin_
↳ unet')
```

```
[ ]:
```



## INDICES AND TABLES

- `genindex`
- `modindex`
- `search`