1.0 Library Imports

Below we import the libraries we will be using in this notebook.

Marker note, references will be made if code was taken from or inspired by a source. The references can be found in the report.

In [1]:
```
%pip install tensorflow
%pip install os
%pip install numpy
%pip install skimage
%pip install sklearn
%pip install matplotlib
%pip install xmltodict
%pip install pix2pix
%pip install git+https://github.com/tensorflow/examples.git
```

```
Requirement already satisfied: tensorflow in /opt/anaconda3/envs/london/lib/python3.11/si
te-packages (2.12.0)
Requirement already satisfied: absl-py>=1.0.0 in /opt/anaconda3/envs/london/lib/python3.1
1/site-packages (from tensorflow) (2.1.0)
Requirement already satisfied: astunparse>=1.6.0 in /opt/anaconda3/envs/london/lib/python
3.11/site-packages (from tensorflow) (1.6.3)
Requirement already satisfied: flatbuffers>=2.0 in /opt/anaconda3/envs/london/lib/python
3.11/site-packages (from tensorflow) (2.0)
Requirement already satisfied: gast<=0.4.0,>=0.2.1 in /opt/anaconda3/envs/london/lib/pyth
on3.11/site-packages (from tensorflow) (0.4.0)
Requirement already satisfied: google-pasta>=0.1.1 in /opt/anaconda3/envs/london/lib/pyth
on3.11/site-packages (from tensorflow) (0.2.0)
Requirement already satisfied: h5py>=2.9.0 in /opt/anaconda3/envs/london/lib/python3.11/s
ite-packages (from tensorflow) (3.11.0)
Requirement already satisfied: jax>=0.3.15 in /opt/anaconda3/envs/london/lib/python3.11/s
ite-packages (from tensorflow) (0.4.30)
Requirement already satisfied: libclang>=13.0.0 in /opt/anaconda3/envs/london/lib/python
3.11/site-packages (from tensorflow) (18.1.1)
Requirement already satisfied: numpy<1.24,>=1.22 in /opt/anaconda3/envs/london/lib/python
3.11/site-packages (from tensorflow) (1.23.5)
Requirement already satisfied: opt-einsum>=2.3.2 in /opt/anaconda3/envs/london/lib/python
3.11/site-packages (from tensorflow) (3.3.0)
Requirement already satisfied: packaging in /opt/anaconda3/envs/london/lib/python3.11/sit
e-packages (from tensorflow) (23.2)
Requirement already satisfied: protobuf!=4.21.0,!=4.21.1,!=4.21.2,!=4.21.3,!=4.21.4,!=4.2
1.5,<5.0.0dev,>=3.20.3 in /opt/anaconda3/envs/london/lib/python3.11/site-packages (from t
ensorflow) (3.20.3)
Requirement already satisfied: setuptools in /opt/anaconda3/envs/london/lib/python3.11/si
te-packages (from tensorflow) (69.5.1)
Requirement already satisfied: six>=1.12.0 in /opt/anaconda3/envs/london/lib/python3.11/s
ite-packages (from tensorflow) (1.16.0)
Requirement already satisfied: termcolor>=1.1.0 in /opt/anaconda3/envs/london/lib/python
3.11/site-packages (from tensorflow) (2.1.0)
Requirement already satisfied: typing-extensions>=3.6.6 in /opt/anaconda3/envs/london/li
b/python3.11/site-packages (from tensorflow) (4.11.0)
Requirement already satisfied: wrapt<1.15,>=1.11.0 in /opt/anaconda3/envs/london/lib/pyth
on3.11/site-packages (from tensorflow) (1.14.1)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /opt/anaconda3/envs/london/lib/pyth
on3.11/site-packages (from tensorflow) (1.48.2)
Requirement already satisfied: tensorboard<2.13,>=2.12 in /opt/anaconda3/envs/london/lib/
python3.11/site-packages (from tensorflow) (2.12.1)
Requirement already satisfied: tensorflow-estimator<2.13,>=2.12.0 in /opt/anaconda3/envs/
london/lib/python3.11/site-packages (from tensorflow) (2.12.0)
Requirement already satisfied: keras<2.13,>=2.12.0 in /opt/anaconda3/envs/london/lib/pyth
on3.11/site-packages (from tensorflow) (2.12.0)
Requirement already satisfied: wheel<1.0,>=0.23.0 in /opt/anaconda3/envs/london/lib/pytho
n3.11/site-packages (from astunparse>=1.6.0->tensorflow) (0.35.1)
Requirement already satisfied: jaxlib<=0.4.30,>=0.4.27 in /opt/anaconda3/envs/london/lib/
python3.11/site-packages (from jax>=0.3.15->tensorflow) (0.4.30)
Requirement already satisfied: ml-dtypes>=0.2.0 in /opt/anaconda3/envs/london/lib/python
3.11/site-packages (from jax>=0.3.15->tensorflow) (0.5.1)
Requirement already satisfied: scipy>=1.9 in /opt/anaconda3/envs/london/lib/python3.11/si
te-packages (from jax>=0.3.15->tensorflow) (1.13.1)
Requirement already satisfied: google-auth<3,>=1.6.3 in /opt/anaconda3/envs/london/lib/py
thon3.11/site-packages (from tensorboard<2.13,>=2.12->tensorflow) (2.29.0)
Requirement already satisfied: google-auth-oauthlib<1.1,>=0.5 in /opt/anaconda3/envs/lond
on/lib/python3.11/site-packages (from tensorboard<2.13,>=2.12->tensorflow) (0.5.2)
Requirement already satisfied: markdown>=2.6.8 in /opt/anaconda3/envs/london/lib/python
3.11/site-packages (from tensorboard<2.13,>=2.12->tensorflow) (3.4.1)
Requirement already satisfied: requests<3,>=2.21.0 in /opt/anaconda3/envs/london/lib/pyth
on3.11/site-packages (from tensorboard<2.13,>=2.12->tensorflow) (2.32.2)
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in /opt/anaconda3/en
```

```
vs/london/lib/python3.11/site-packages (from tensorboard<2.13,>=2.12->tensorflow) (0.7.0)
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in /opt/anaconda3/envs/londo
n/lib/python3.11/site-packages (from tensorboard<2.13,>=2.12->tensorflow) (1.8.1)
Requirement already satisfied: werkzeug>=1.0.1 in /opt/anaconda3/envs/london/lib/python
3.11/site-packages (from tensorboard<2.13,>=2.12->tensorflow) (3.0.3)
Requirement already satisfied: cachetools<6.0,>=2.0.0 in /opt/anaconda3/envs/london/lib/p
ython3.11/site-packages (from google-auth<3,>=1.6.3->tensorboard<2.13,>=2.12->tensorflow)
(5.3.3)
Requirement already satisfied: pyasn1-modules>=0.2.1 in /opt/anaconda3/envs/london/lib/py
thon3.11/site-packages (from google-auth<3,>=1.6.3->tensorboard<2.13,>=2.12->tensorflow)
(0.2.8)
Requirement already satisfied: rsa<5,>=3.1.4 in /opt/anaconda3/envs/london/lib/python3.1
1/site-packages (from google-auth<3,>=1.6.3->tensorboard<2.13,>=2.12->tensorflow) (4.7.2)
Requirement already satisfied: requests-oauthlib>=0.7.0 in /opt/anaconda3/envs/london/li
b/python3.11/site-packages (from google-auth-oauthlib<1.1,>=0.5->tensorboard<2.13,>=2.12-
>tensorflow) (1.3.0)
Requirement already satisfied: charset-normalizer<4,>=2 in /opt/anaconda3/envs/london/li
b/python3.11/site-packages (from requests<3,>=2.21.0->tensorboard<2.13,>=2.12->tensorflo
w) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in /opt/anaconda3/envs/london/lib/python3.11/
site-packages (from requests<3,>=2.21.0->tensorboard<2.13,>=2.12->tensorflow) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in /opt/anaconda3/envs/london/lib/pytho
n3.11/site-packages (from requests<3,>=2.21.0->tensorboard<2.13,>=2.12->tensorflow)
(2.2.1)
Requirement already satisfied: certifi>=2017.4.17 in /opt/anaconda3/envs/london/lib/pytho
n3.11/site-packages (from requests<3,>=2.21.0->tensorboard<2.13,>=2.12->tensorflow) (202
5.1.31)
Requirement already satisfied: MarkupSafe>=2.1.1 in /opt/anaconda3/envs/london/lib/python
3.11/site-packages (from werkzeug>=1.0.1->tensorboard<2.13,>=2.12->tensorflow) (2.1.3)
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in /opt/anaconda3/envs/london/lib/pyt
hon3.11/site-packages (from pyasn1-modules>=0.2.1->google-auth<3,>=1.6.3->tensorboard<2.1
3,>=2.12->tensorflow) (0.4.8)
Requirement already satisfied: oauthlib>=3.0.0 in /opt/anaconda3/envs/london/lib/python
3.11/site-packages (from requests-oauthlib>=0.7.0->google-auth-oauthlib<1.1,>=0.5->tensor
board<2.13,>=2.12->tensorflow) (3.2.2)
Note: you may need to restart the kernel to use updated packages.
ERROR: Could not find a version that satisfies the requirement os (from versions: none)
ERROR: No matching distribution found for os
Note: you may need to restart the kernel to use updated packages.
Requirement already satisfied: numpy in /opt/anaconda3/envs/london/lib/python3.11/site-pa
ckages (1.23.5)
Note: you may need to restart the kernel to use updated packages.
Collecting skimage
  Using cached skimage-0.0.tar.gz (757 bytes)
  Preparing metadata (setup.py) ... error
  error: subprocess-exited-with-error

  × python setup.py egg_info did not run successfully.
  │ exit code: 1
  └─> [3 lines of output]

      *** Please install the `scikit-image` package (instead of `skimage`) ***

      [end of output]

  note: This error originates from a subprocess, and is likely not a problem with pip.
error: metadata-generation-failed

× Encountered error while generating package metadata.
└─> See above for output.

note: This is an issue with the package mentioned above, not pip.
```

```
hint: See above for details.
Note: you may need to restart the kernel to use updated packages.
Collecting sklearn
  Using cached sklearn-0.0.post12.tar.gz (2.6 kB)
  Preparing metadata (setup.py) ... error
  error: subprocess-exited-with-error

  × python setup.py egg_info did not run successfully.
  │ exit code: 1
  ╰─> [15 lines of output]
      The 'sklearn' PyPI package is deprecated, use 'scikit-learn'
      rather than 'sklearn' for pip commands.

      Here is how to fix this error in the main use cases:
      - use 'pip install scikit-learn' rather than 'pip install sklearn'
      - replace 'sklearn' by 'scikit-learn' in your pip requirements files
        (requirements.txt, setup.py, setup.cfg, Pipfile, etc ...)
      - if the 'sklearn' package is used by one of your dependencies,
        it would be great if you take some time to track which package uses
        'sklearn' instead of 'scikit-learn' and report it to their issue tracker
      - as a last resort, set the environment variable
        SKLEARN_ALLOW_DEPRECATED_SKLEARN_PACKAGE_INSTALL=True to avoid this error

      More information is available at
      https://github.com/scikit-learn/sklearn-pypi-package
      [end of output]

  note: This error originates from a subprocess, and is likely not a problem with pip.
error: metadata-generation-failed

× Encountered error while generating package metadata.
╰─> See above for output.

note: This is an issue with the package mentioned above, not pip.
hint: See above for details.
Note: you may need to restart the kernel to use updated packages.
Requirement already satisfied: matplotlib in /opt/anaconda3/envs/london/lib/python3.11/si
te-packages (3.8.4)
Requirement already satisfied: contourpy>=1.0.1 in /opt/anaconda3/envs/london/lib/python
3.11/site-packages (from matplotlib) (1.2.0)
Requirement already satisfied: cycler>=0.10 in /opt/anaconda3/envs/london/lib/python3.11/
site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in /opt/anaconda3/envs/london/lib/python
3.11/site-packages (from matplotlib) (4.51.0)
Requirement already satisfied: kiwisolver>=1.3.1 in /opt/anaconda3/envs/london/lib/python
3.11/site-packages (from matplotlib) (1.4.4)
Requirement already satisfied: numpy>=1.21 in /opt/anaconda3/envs/london/lib/python3.11/s
ite-packages (from matplotlib) (1.23.5)
Requirement already satisfied: packaging>=20.0 in /opt/anaconda3/envs/london/lib/python
3.11/site-packages (from matplotlib) (23.2)
Requirement already satisfied: pillow>=8 in /opt/anaconda3/envs/london/lib/python3.11/sit
e-packages (from matplotlib) (10.3.0)
Requirement already satisfied: pyparsing>=2.3.1 in /opt/anaconda3/envs/london/lib/python
3.11/site-packages (from matplotlib) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in /opt/anaconda3/envs/london/lib/pyt
hon3.11/site-packages (from matplotlib) (2.9.0.post0)
Requirement already satisfied: six>=1.5 in /opt/anaconda3/envs/london/lib/python3.11/sit
e-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
Requirement already satisfied: xmltodict in /opt/anaconda3/envs/london/lib/python3.11/sit
e-packages (0.13.0)
Note: you may need to restart the kernel to use updated packages.
```

```
ERROR: Could not find a version that satisfies the requirement pix2pix (from versions: no
ne)
ERROR: No matching distribution found for pix2pix
Note: you may need to restart the kernel to use updated packages.
Collecting git+https://github.com/tensorflow/examples.git
  Cloning https://github.com/tensorflow/examples.git to /private/var/folders/r4/mvngz2z96
txg8fqvpmt13pr40000gn/T/pip-req-build-ge_1uicb
  Running command git clone --filter=blob:none --quiet https://github.com/tensorflow/exam
ples.git /private/var/folders/r4/mvngz2z96txg8fqvpmt13pr40000gn/T/pip-req-build-ge_1uicb
  Resolved https://github.com/tensorflow/examples.git to commit fed63294c10c71b7da028e0e7
5de9ff68ac56d17
  Preparing metadata (setup.py) ... done
Requirement already satisfied: absl-py in /opt/anaconda3/envs/london/lib/python3.11/site-
packages (from tensorflow-examples==0.1741282361.145486042149443330204603306743204048930
0359736599) (2.1.0)
Requirement already satisfied: six in /opt/anaconda3/envs/london/lib/python3.11/site-pack
ages (from tensorflow-examples==0.1741282361.145486042149443330204603306743204048930035
97 36599) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

In [2]:
```python
# Tensorflow imports.
import tensorflow as tf
from tensorflow.keras.layers import Input, Conv2D, MaxPooling2D, UpSampling2D, concatena
from tensorflow_examples.models.pix2pix import pix2pix

# OS import to access files.
import os

# Numpy Import for array normalizing.
import numpy as np

# skimage import for photo processing.
import skimage as ski
import sklearn as skl
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

# matplot import for plotting.
import matplotlib.pyplot as plt
from matplotlib.patches import Polygon
from matplotlib.collections import PatchCollection
import seaborn as sns

# xmltodict import is used to parse the annotation file.
import xmltodict
```

2.0 Load Dataset with Annotations

Below we need to load in the dataset with the corresponding annotations. We will also be resizing the dataset to 512x512 images to ensure we have consistency for loading in the neural networks.

2.1 Define resize_points function

In [3]:
```python
def resize_points(polygons, width, height):
    """
    Resizes polygon point coordinates to fit a 512x512 image resolution.
```

```
    Parameters:
        polygons (list of dict): A list of polygon annotations, each containing:
            - '@label' (str): The polygon's name (e.g., 'Post', 'Sign').
            - '@points' (str): A semicolon-separated string of x,y coordinates.
        width (int): The original image width.
        height (int): The original image height.

    Returns:
        dict: A dictionary with labels as keys ('Post', 'Sign', etc.) and their correspo
    """
    # Create a container and set zero values for potential missing items.
    resized_polygons = {'Post': None, # Class 1
                        'Sign': None, # Class 2
                        'Lower Plate': None, # Class 3
                        'Top Plate': None} # Class 4

    # Loop through each polygon.
    for polygon in polygons:
        # Run within a try block as some odd exporting polygons exist in the annotation
        try:
            # Create storage for the related points in this polygon.
            points = polygon['@points']

            # Get the image scale based on the x and y factors.
            scale_x = 128 / width
            scale_y = 128 / height

            # Parse the polygon points.
            points = points.split(';')
            # Split the points and convert to float format. Reference 22 helped with thi
            points = np.array([list(map(float, point.split(','))) for point in points])

            # Scale the points to the new resolution.
            scaled_points = []
            for point in points:
                scaled_points.append([point[0] * scale_x, point[1] * scale_y])

            # Store the scaled points in the container.
            resized_polygons[polygon['@label']] = np.array(scaled_points)
        except:
            print('Invalid polygon provided', polygon, '\n')

    # Return stored points
    return resized_polygons
```

2.2 Define get_mask function

In [4]:

```
def get_mask(points):
    """
    Generates a segmentation mask from polygon annotations.

    Parameters:
        points (dict): A dictionary where keys are object classes ('Post', 'Sign', 'Lowe
        If a value is None, it is skipped.

    Returns:
        np.ndarray: A 128x128 numpy array mask where:
                    0 = background
                    1 = post
                    2 = sign
```

```
                    3 = lower plate
                    4 = top plate
    """
    # Create mask array.
    mask = np.zeros((128, 128), dtype = np.uint8)

    for key in points:
        polygon = points[key]
        # Continue if polygon is None.
        if polygon is None:
            continue

        # Get mask based on polygon.
        # Code from references 23 and 25. 25 was instrumental in getting this properly s
        rr, cc = ski.draw.polygon(polygon[:, 1], polygon[:, 0], mask.shape)

        # Overlay mask with pixel value depending on the class of polygon.
        # Class 1 = post
        # Class 2 = sign
        # Class 3 = Lower Plate
        # Class 4 = Top Plate
        if key == 'Post':
            mask[rr, cc] = 1
        elif key == 'Sign':
            mask[rr, cc] = 2
        elif key == 'Lower Plate':
            mask[rr, cc] = 3
        elif key == 'Top Plate':
            mask[rr, cc] = 4

    # Return the mask
    return mask
```

2.3 Define normalize function to store image as tensor.

In [5]:
```
def normalize(input_image):
    """
    Normalizes an image's pixel values to the range [0, 1].

    Parameters:
        input_image (tf.Tensor or np.ndarray): The input image with pixel values in the

    Returns:
        tf.Tensor: The normalized image with pixel values scaled to [0, 1].
    """
    # Normalize the pixel range values between [0:1]
    # Taken from reference 25.
    image = tf.cast(input_image, dtype = tf.float32) / 255.0
    return image
```

2.3 Load and Resize the Images/Masks

Below we will load in the images, resize them and load in and reszie the associated masks. This essentially sets up the dataset for use in machine learning algorithms.

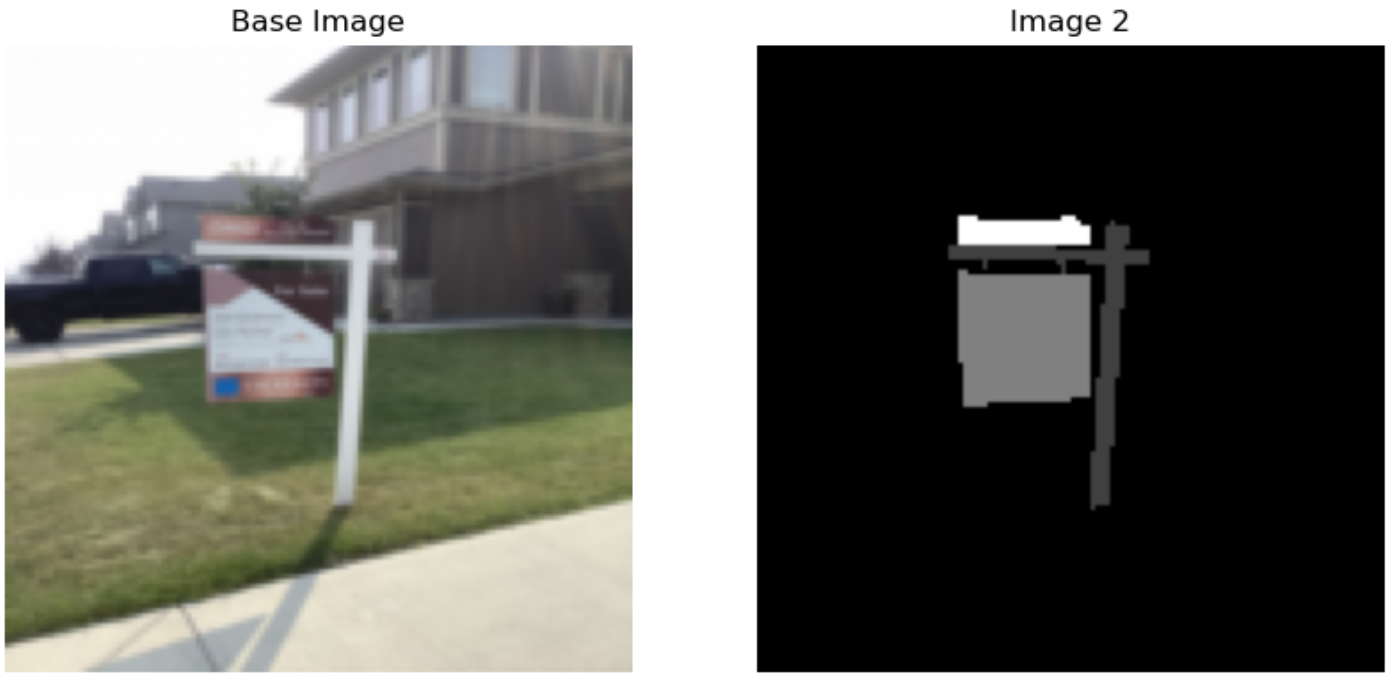
In [6]:
```
# Create a storage containers.
dataset = {}
annotations = []
```

```python
# Parse the XML file
# Code reference: [15]
with open("./object_detection/annotations.xml", "r") as f:
    # Import the annotations in a dictionary format.
    annotations = xmltodict.parse(f.read())

# Loop through every image object in the annotations and store the image.
for image in annotations["annotations"]["image"]:
    # Get image name.
    image_name = image["@name"]

    # Load the image with scikit learn.
    photo = ski.io.imread("./object_detection/annotated_images" + '/' + image_name)

    # Create a resized image (224x224) in the storage container.
    # We'll need to store the images in 224x224 to standardize them for the neural netwo
    resized_image = ski.transform.resize(photo, (128, 128), anti_aliasing=True)

    # Resize and standardize the polygon points
    resized_points = resize_points(image['polygon'], photo.shape[1], photo.shape[0])

    # Get the mask.
    mask = get_mask(resized_points)

    # Store the resized image and all associated masks.
    dataset[image_name] = {'image': resized_image,
                           'tensor': normalize(resized_image),
                           'mask': mask}
```

```
Invalid polygon provided @label

Invalid polygon provided @source

Invalid polygon provided @occluded

Invalid polygon provided @points

Invalid polygon provided @z_order

Invalid polygon provided @label

Invalid polygon provided @source

Invalid polygon provided @occluded

Invalid polygon provided @points

Invalid polygon provided @z_order

Invalid polygon provided @label

Invalid polygon provided @source

Invalid polygon provided @occluded

Invalid polygon provided @points

Invalid polygon provided @z_order
```

2.4 Confirm Masks Conform to Resized Image

In [7]:
```
# Choose a random image
image_info = dataset['f_1723741777.jpg']

# Load photos.
photo = image_info['image']
photo_mask = photo.copy()

# Layer masks over photo.
mask = image_info['mask']

# Create plot.
fig, axes = plt.subplots(1, 2, figsize = (10, 5))

# Show base image
axes[0].imshow(photo, cmap='gray')
axes[0].set_title("Base Image")
axes[0].axis('off')

# Show image with mask.
axes[1].imshow(mask, cmap='gray')
axes[1].set_title("Image 2")
axes[1].axis('off')

# Show plot.
plt.show()
```



Above we can see the mask does align with the corresponding image and highlights the items we want to detect in a pixel-wise manner.

2.5 Convert data to Tensors Sets

In [8]:
```
# Convert dictionary values to lists
images = [v['tensor'] for v in dataset.values()]
masks = [v['mask'] for v in dataset.values()]
```

```
# Convert to numpy arrays
images = np.array(images, dtype=np.float32)
masks = np.array(masks, dtype=np.int32)

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(images, masks, test_size=0.1, random
```

In [9]:
```
BATCH_SIZE = 32
# Get the train dataset. We set it to repeat due to the small size of the dataset.
train_ds = tf.data.Dataset.from_tensor_slices((X_train, y_train)).shuffle(1000).batch(BA

# Get the test dataset.
test_ds = tf.data.Dataset.from_tensor_slices((X_test, y_test)).batch(BATCH_SIZE).repeat(

# Set
EPOCHS = 20
steps_per_epoch = len(X_train) // BATCH_SIZE
validation_steps = len(X_test) // BATCH_SIZE
```

3.0 Set up Base Model Architecture

Below we will set up the base model and run it individually on the different masks. Most of the code below was heavily inspired or taken from reference 23.

In [10]:
```
# Load in the base model.
base_model = tf.keras.applications.MobileNetV2(input_shape=[128, 128, 3], include_top=Fa

# Use the activations of these layers
layer_names = [
    'block_1_expand_relu',   # 64x64
    'block_3_expand_relu',   # 32x32
    'block_6_expand_relu',   # 16x16
    'block_13_expand_relu',  # 8x8
    'block_16_project',      # 4x4
]

base_model_outputs = [base_model.get_layer(name).output for name in layer_names]

# Create the feature extraction model
down_stack = tf.keras.Model(inputs=base_model.input, outputs=base_model_outputs)

down_stack.trainable = False
```

In [11]:
```
# Load in stacks for decoder.
up_stack = [
    pix2pix.upsample(512, 3),  # 4x4 -> 8x8
    pix2pix.upsample(256, 3),  # 8x8 -> 16x16
    pix2pix.upsample(128, 3),  # 16x16 -> 32x32
    pix2pix.upsample(64, 3),   # 32x32 -> 64x64
]
```

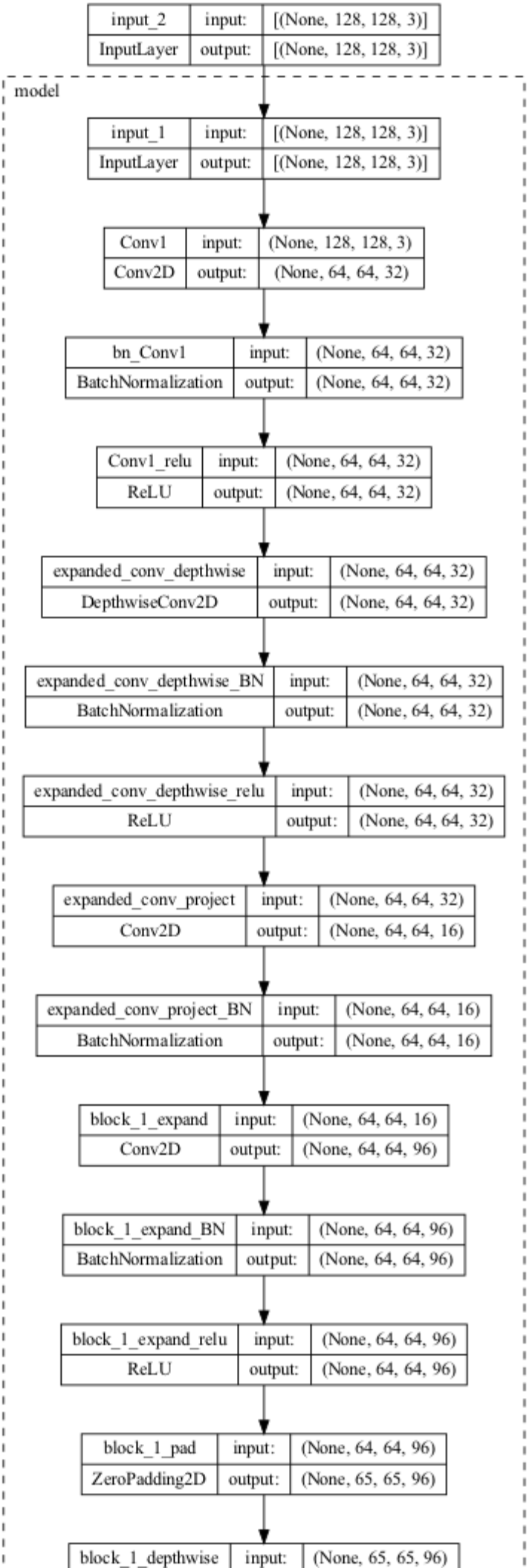
In [12]:
```
# Model code was taken directly from reference 27.
def unet_model(output_channels:int):
  inputs = tf.keras.layers.Input(shape=[128, 128, 3])

  # Downsampling through the model
  skips = down_stack(inputs)
  x = skips[-1]
```

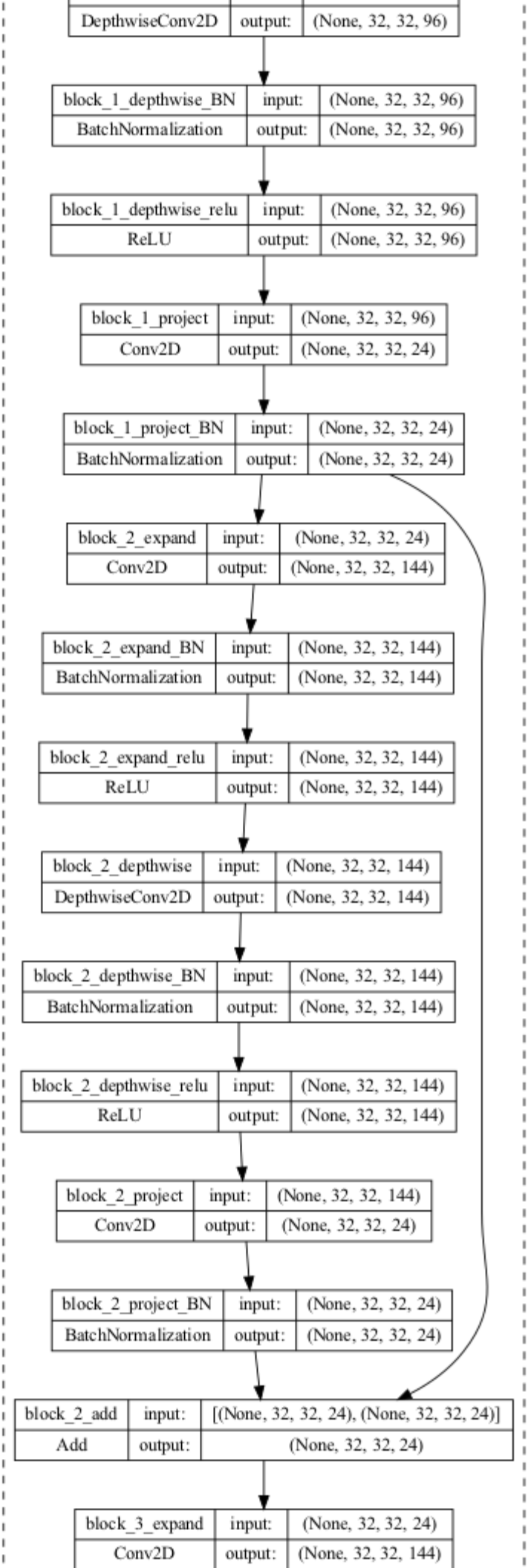
```
  skips = reversed(skips[:-1])

  # Upsampling and establishing the skip connections
  for up, skip in zip(up_stack, skips):
    x = up(x)
    concat = tf.keras.layers.Concatenate()
    x = concat([x, skip])

  # This is the last layer of the model
  last = tf.keras.layers.Conv2DTranspose(
      filters=output_channels, kernel_size=3, strides=2,
      padding='same')  #64x64 -> 128x128

  x = last(x)

  return tf.keras.Model(inputs=inputs, outputs=x)
```

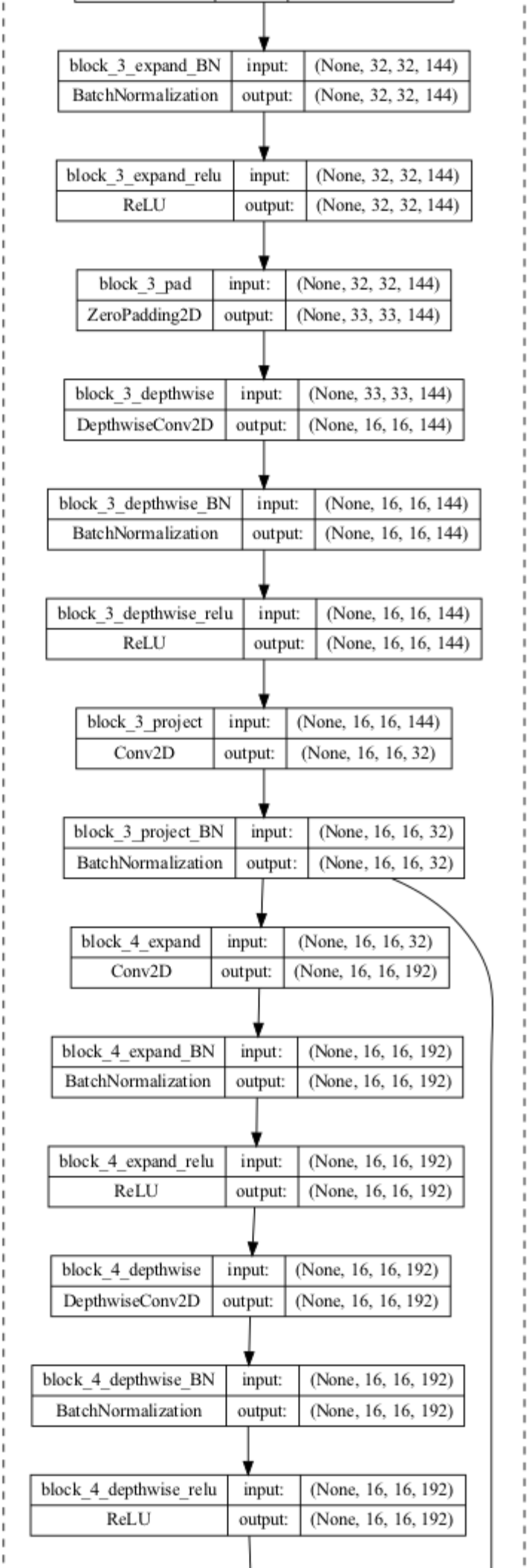
In [13]:

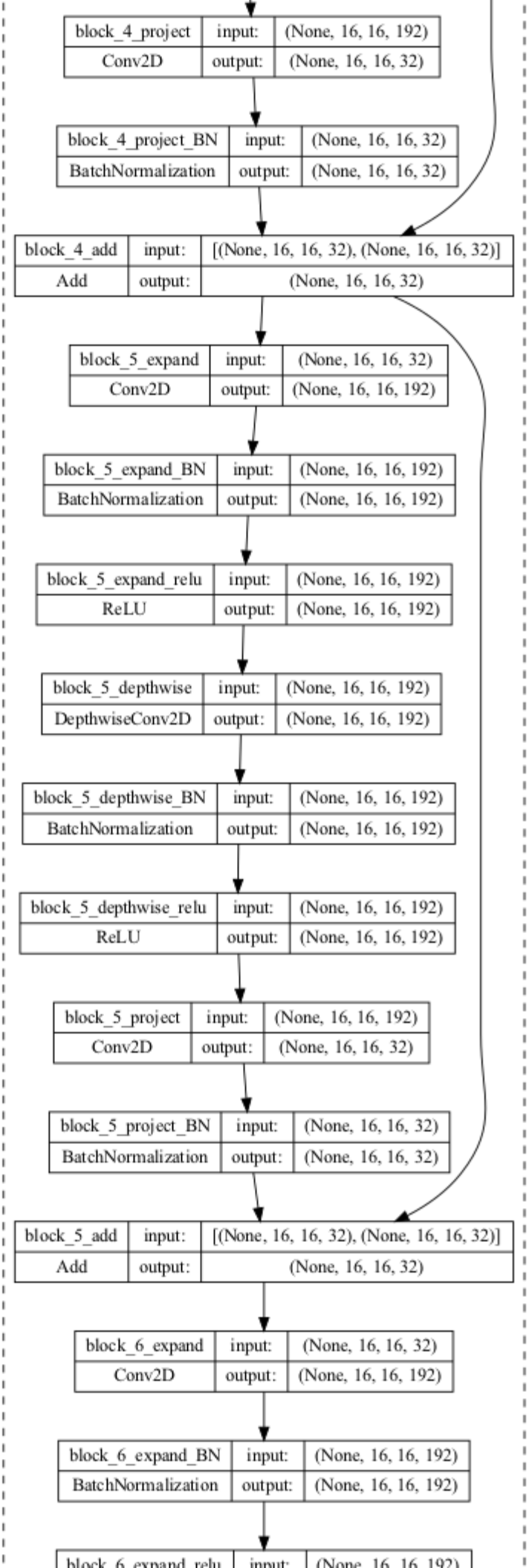
```
# Compile the model.
model = unet_model(output_channels=5)
model.compile(optimizer='adam', loss=tf.keras.losses.SparseCategoricalCrossentropy(from_

# Plot the model.
tf.keras.utils.plot_model(model, show_shapes=True, expand_nested=True, dpi=64)
```

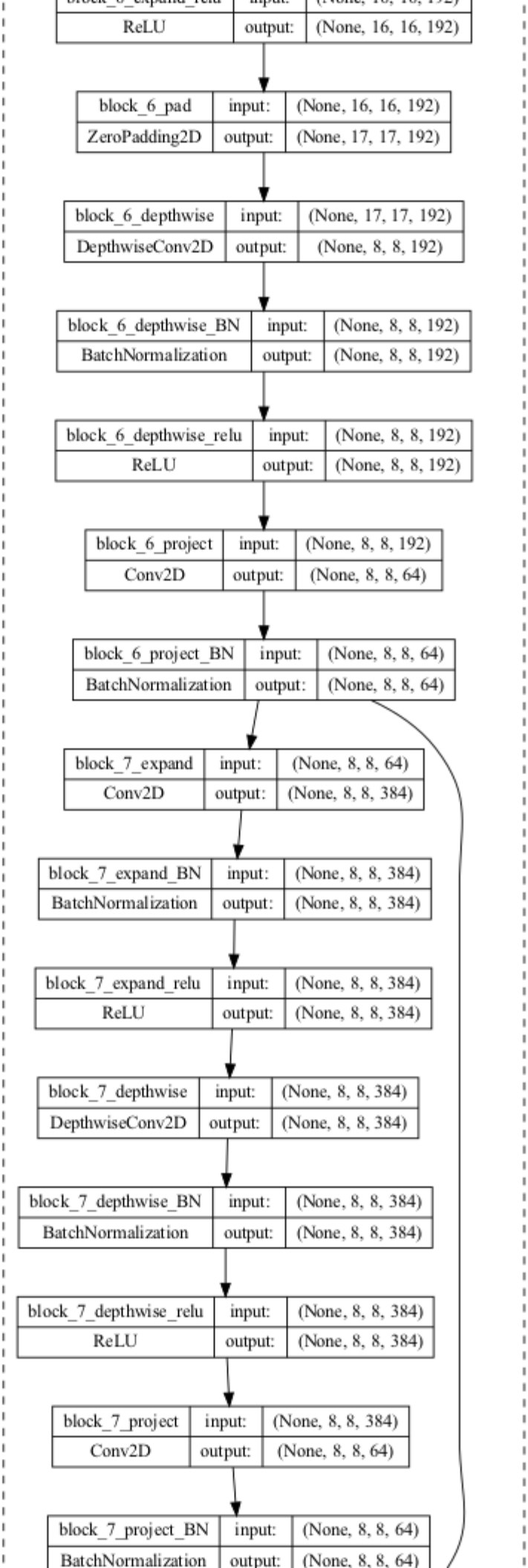
Out[13]:

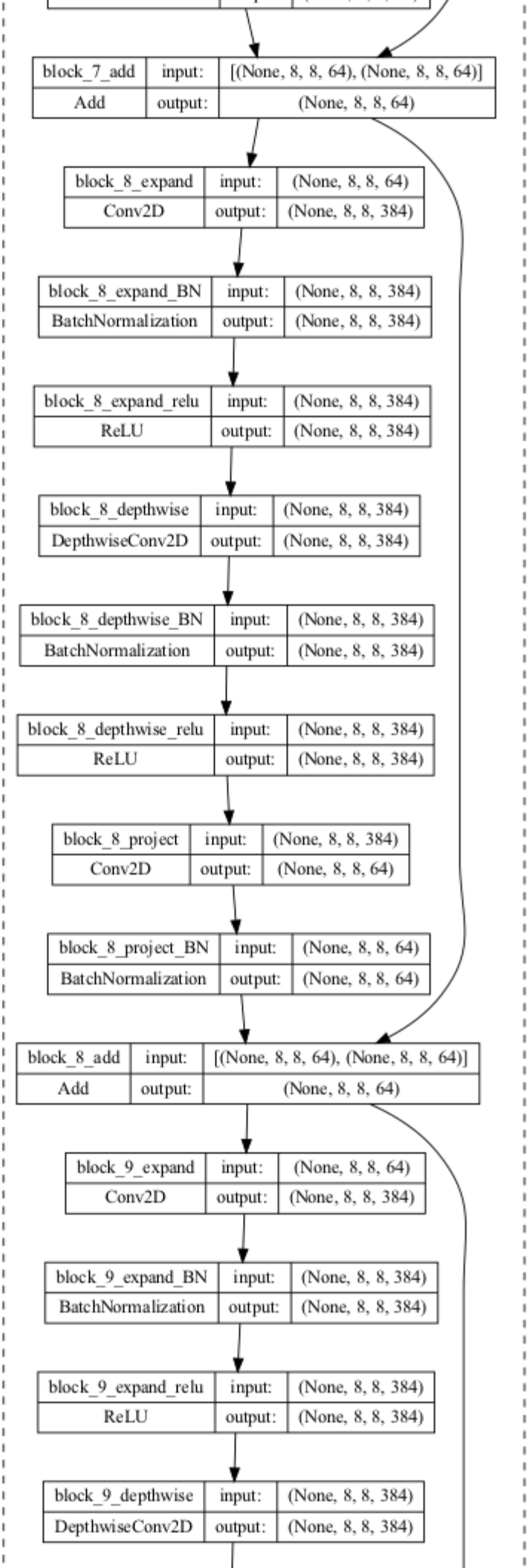

input_2 | input: | [(None, 128, 128, 3)]
InputLayer | output: | [(None, 128, 128, 3)]
model
input_1 | input: | [(None, 128, 128, 3)]
InputLayer | output: | [(None, 128, 128, 3)]
Conv1 | input: | (None, 128, 128, 3)
Conv2D | output: | (None, 64, 64, 32)
bn_Conv1 | input: | (None, 64, 64, 32)
BatchNormalization | output: | (None, 64, 64, 32)
Conv1_relu | input: | (None, 64, 64, 32)
ReLU | output: | (None, 64, 64, 32)
expanded_conv_depthwise | input: | (None, 64, 64, 32)
DepthwiseConv2D | output: | (None, 64, 64, 32)
expanded_conv_depthwise_BN | input: | (None, 64, 64, 32)
BatchNormalization | output: | (None, 64, 64, 32)
expanded_conv_depthwise_relu | input: | (None, 64, 64, 32)
ReLU | output: | (None, 64, 64, 32)
expanded_conv_project | input: | (None, 64, 64, 32)
Conv2D | output: | (None, 64, 64, 16)
expanded_conv_project_BN | input: | (None, 64, 64, 16)
BatchNormalization | output: | (None, 64, 64, 16)
block_1_expand | input: | (None, 64, 64, 16)
Conv2D | output: | (None, 64, 64, 96)
block_1_expand_BN | input: | (None, 64, 64, 96)
BatchNormalization | output: | (None, 64, 64, 96)
block_1_expand_relu | input: | (None, 64, 64, 96)
ReLU | output: | (None, 64, 64, 96)
block_1_pad | input: | (None, 64, 64, 96)
ZeroPadding2D | output: | (None, 65, 65, 96)
block_1_depthwise | input: | (None, 65, 65, 96)

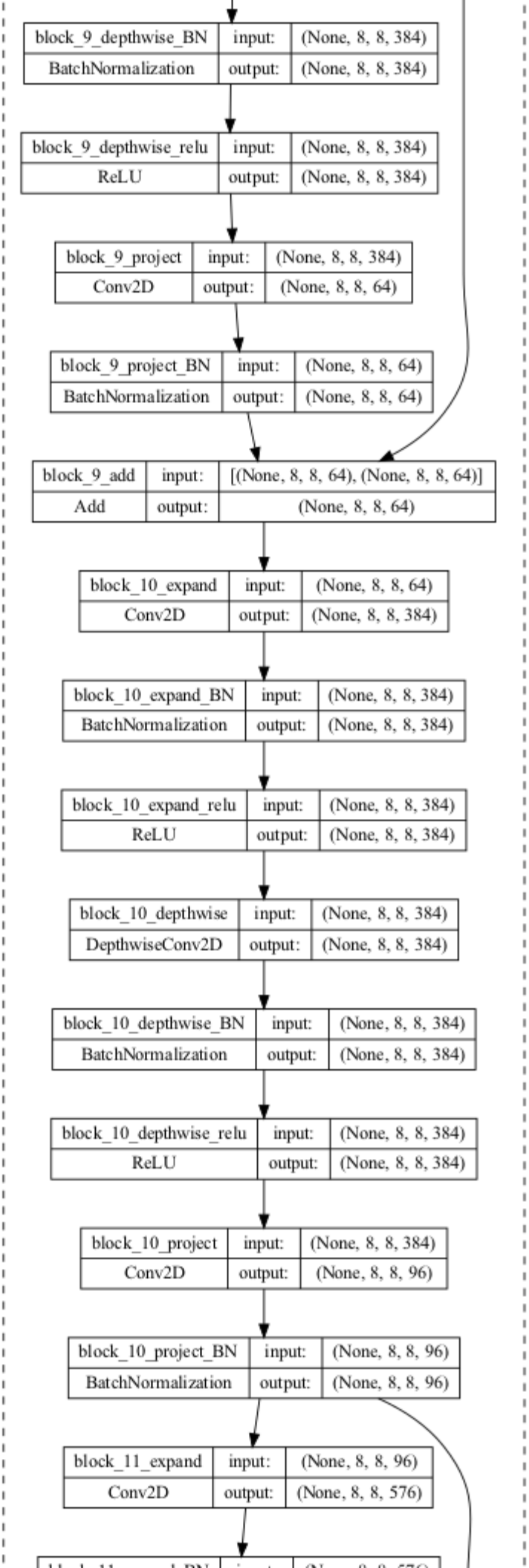

DepthwiseConv2D | output: | (None, 32, 32, 96)
block_1_depthwise_BN | input: | (None, 32, 32, 96)
BatchNormalization | output: | (None, 32, 32, 96)
block_1_depthwise_relu | input: | (None, 32, 32, 96)
ReLU | output: | (None, 32, 32, 96)
block_1_project | input: | (None, 32, 32, 96)
Conv2D | output: | (None, 32, 32, 24)
block_1_project_BN | input: | (None, 32, 32, 24)
BatchNormalization | output: | (None, 32, 32, 24)
block_2_expand | input: | (None, 32, 32, 24)
Conv2D | output: | (None, 32, 32, 144)
block_2_expand_BN | input: | (None, 32, 32, 144)
BatchNormalization | output: | (None, 32, 32, 144)
block_2_expand_relu | input: | (None, 32, 32, 144)
ReLU | output: | (None, 32, 32, 144)
block_2_depthwise | input: | (None, 32, 32, 144)
DepthwiseConv2D | output: | (None, 32, 32, 144)
block_2_depthwise_BN | input: | (None, 32, 32, 144)
BatchNormalization | output: | (None, 32, 32, 144)
block_2_depthwise_relu | input: | (None, 32, 32, 144)
ReLU | output: | (None, 32, 32, 144)
block_2_project | input: | (None, 32, 32, 144)
Conv2D | output: | (None, 32, 32, 24)
block_2_project_BN | input: | (None, 32, 32, 24)
BatchNormalization | output: | (None, 32, 32, 24)
block_2_add | input: | [(None, 32, 32, 24), (None, 32, 32, 24)]
Add | output: | (None, 32, 32, 24)
block_3_expand | input: | (None, 32, 32, 24)
Conv2D | output: | (None, 32, 32, 144)

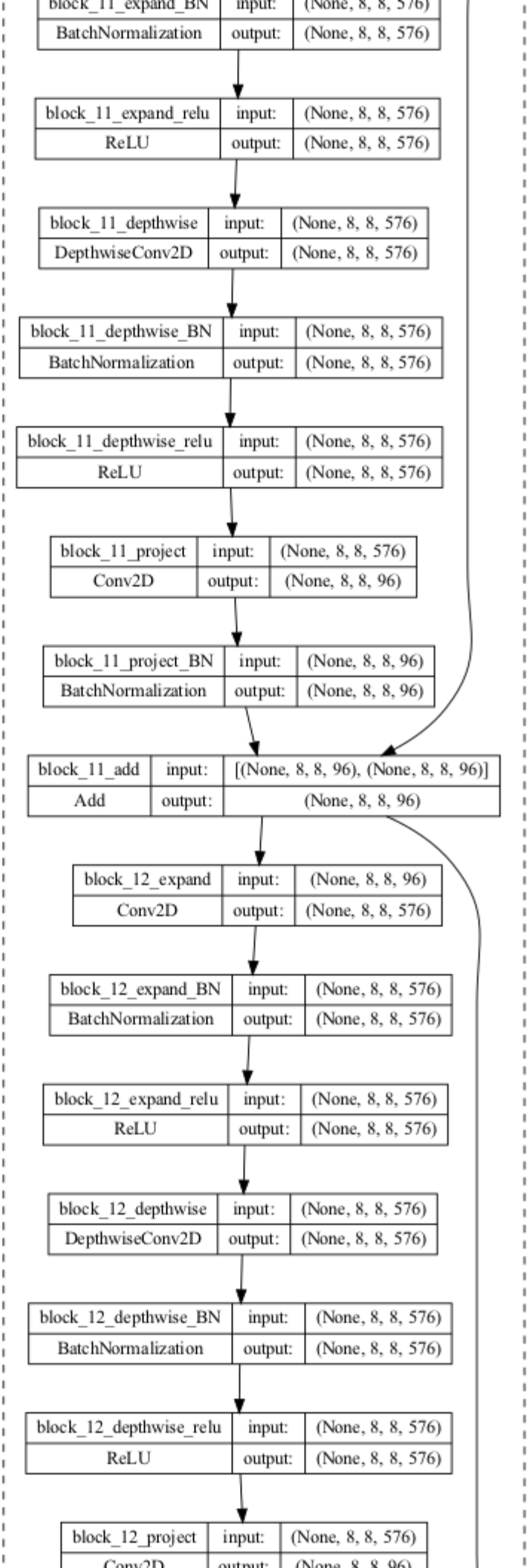

block_3_expand_BN | input: | (None, 32, 32, 144)
BatchNormalization | output: | (None, 32, 32, 144)
block_3_expand_relu | input: | (None, 32, 32, 144)
ReLU | output: | (None, 32, 32, 144)
block_3_pad | input: | (None, 32, 32, 144)
ZeroPadding2D | output: | (None, 33, 33, 144)
block_3_depthwise | input: | (None, 33, 33, 144)
DepthwiseConv2D | output: | (None, 16, 16, 144)
block_3_depthwise_BN | input: | (None, 16, 16, 144)
BatchNormalization | output: | (None, 16, 16, 144)
block_3_depthwise_relu | input: | (None, 16, 16, 144)
ReLU | output: | (None, 16, 16, 144)
block_3_project | input: | (None, 16, 16, 144)
Conv2D | output: | (None, 16, 16, 32)
block_3_project_BN | input: | (None, 16, 16, 32)
BatchNormalization | output: | (None, 16, 16, 32)
block_4_expand | input: | (None, 16, 16, 32)
Conv2D | output: | (None, 16, 16, 192)
block_4_expand_BN | input: | (None, 16, 16, 192)
BatchNormalization | output: | (None, 16, 16, 192)
block_4_expand_relu | input: | (None, 16, 16, 192)
ReLU | output: | (None, 16, 16, 192)
block_4_depthwise | input: | (None, 16, 16, 192)
DepthwiseConv2D | output: | (None, 16, 16, 192)
block_4_depthwise_BN | input: | (None, 16, 16, 192)
BatchNormalization | output: | (None, 16, 16, 192)
block_4_depthwise_relu | input: | (None, 16, 16, 192)
ReLU | output: | (None, 16, 16, 192)

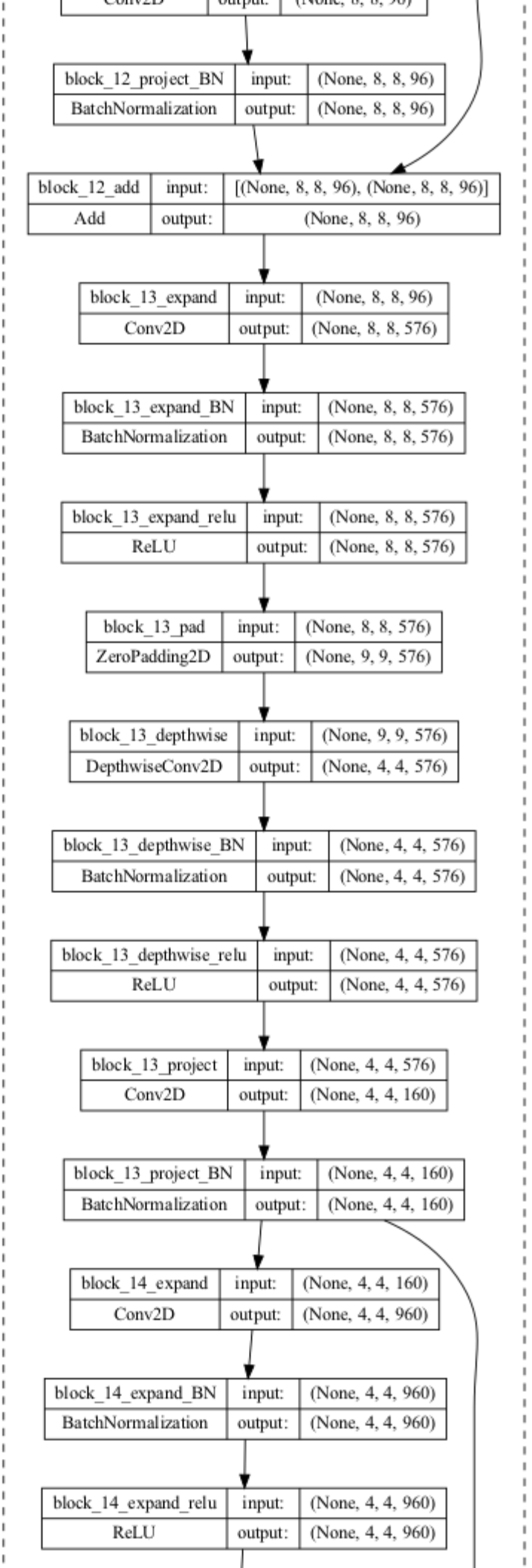

block_4_project | input: | (None, 16, 16, 192)
Conv2D | output: | (None, 16, 16, 32)
block_4_project_BN | input: | (None, 16, 16, 32)
BatchNormalization | output: | (None, 16, 16, 32)
block_4_add | input: | [(None, 16, 16, 32), (None, 16, 16, 32)]
Add | output: | (None, 16, 16, 32)
block_5_expand | input: | (None, 16, 16, 32)
Conv2D | output: | (None, 16, 16, 192)
block_5_expand_BN | input: | (None, 16, 16, 192)
BatchNormalization | output: | (None, 16, 16, 192)
block_5_expand_relu | input: | (None, 16, 16, 192)
ReLU | output: | (None, 16, 16, 192)
block_5_depthwise | input: | (None, 16, 16, 192)
DepthwiseConv2D | output: | (None, 16, 16, 192)
block_5_depthwise_BN | input: | (None, 16, 16, 192)
BatchNormalization | output: | (None, 16, 16, 192)
block_5_depthwise_relu | input: | (None, 16, 16, 192)
ReLU | output: | (None, 16, 16, 192)
block_5_project | input: | (None, 16, 16, 192)
Conv2D | output: | (None, 16, 16, 32)
block_5_project_BN | input: | (None, 16, 16, 32)
BatchNormalization | output: | (None, 16, 16, 32)
block_5_add | input: | [(None, 16, 16, 32), (None, 16, 16, 32)]
Add | output: | (None, 16, 16, 32)
block_6_expand | input: | (None, 16, 16, 32)
Conv2D | output: | (None, 16, 16, 192)
block_6_expand_BN | input: | (None, 16, 16, 192)
BatchNormalization | output: | (None, 16, 16, 192)

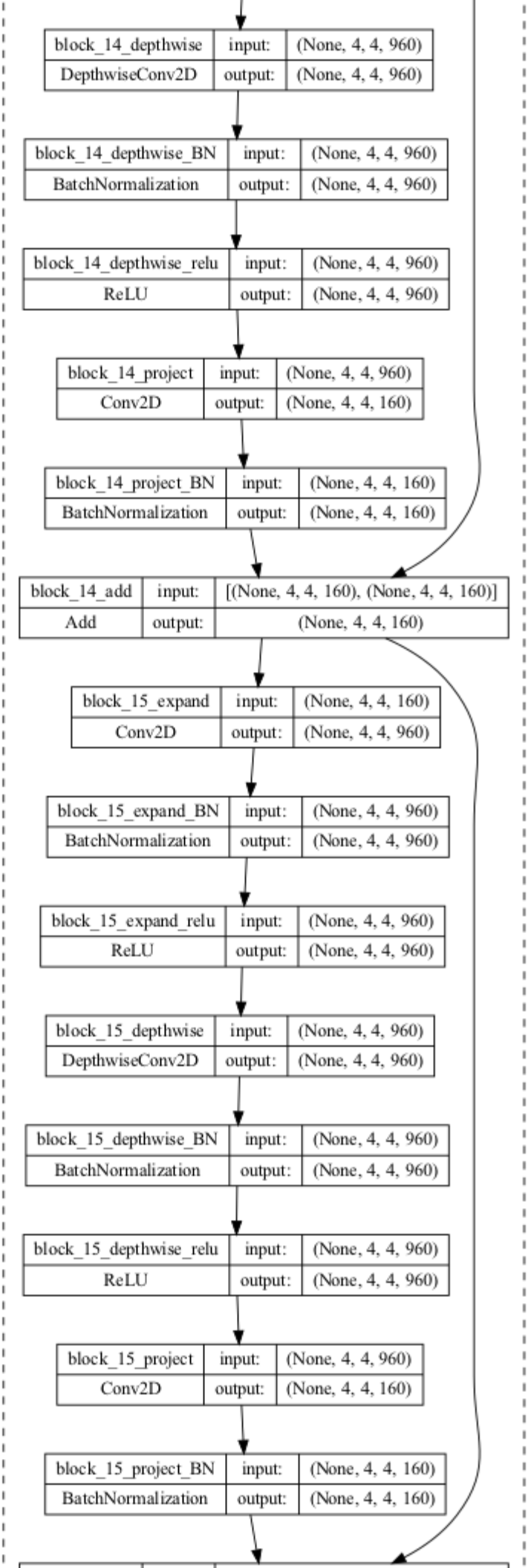

ReLU output: (None, 16, 16, 192)
block_6_pad input: (None, 16, 16, 192)
ZeroPadding2D output: (None, 17, 17, 192)
block_6_depthwise input: (None, 17, 17, 192)
DepthwiseConv2D output: (None, 8, 8, 192)
block_6_depthwise_BN input: (None, 8, 8, 192)
BatchNormalization output: (None, 8, 8, 192)
block_6_depthwise_relu input: (None, 8, 8, 192)
ReLU output: (None, 8, 8, 192)
block_6_project input: (None, 8, 8, 192)
Conv2D output: (None, 8, 8, 64)
block_6_project_BN input: (None, 8, 8, 64)
BatchNormalization output: (None, 8, 8, 64)
block_7_expand input: (None, 8, 8, 64)
Conv2D output: (None, 8, 8, 384)
block_7_expand_BN input: (None, 8, 8, 384)
BatchNormalization output: (None, 8, 8, 384)
block_7_expand_relu input: (None, 8, 8, 384)
ReLU output: (None, 8, 8, 384)
block_7_depthwise input: (None, 8, 8, 384)
DepthwiseConv2D output: (None, 8, 8, 384)
block_7_depthwise_BN input: (None, 8, 8, 384)
BatchNormalization output: (None, 8, 8, 384)
block_7_depthwise_relu input: (None, 8, 8, 384)
ReLU output: (None, 8, 8, 384)
block_7_project input: (None, 8, 8, 384)
Conv2D output: (None, 8, 8, 64)
block_7_project_BN input: (None, 8, 8, 64)
BatchNormalization output: (None, 8, 8, 64)

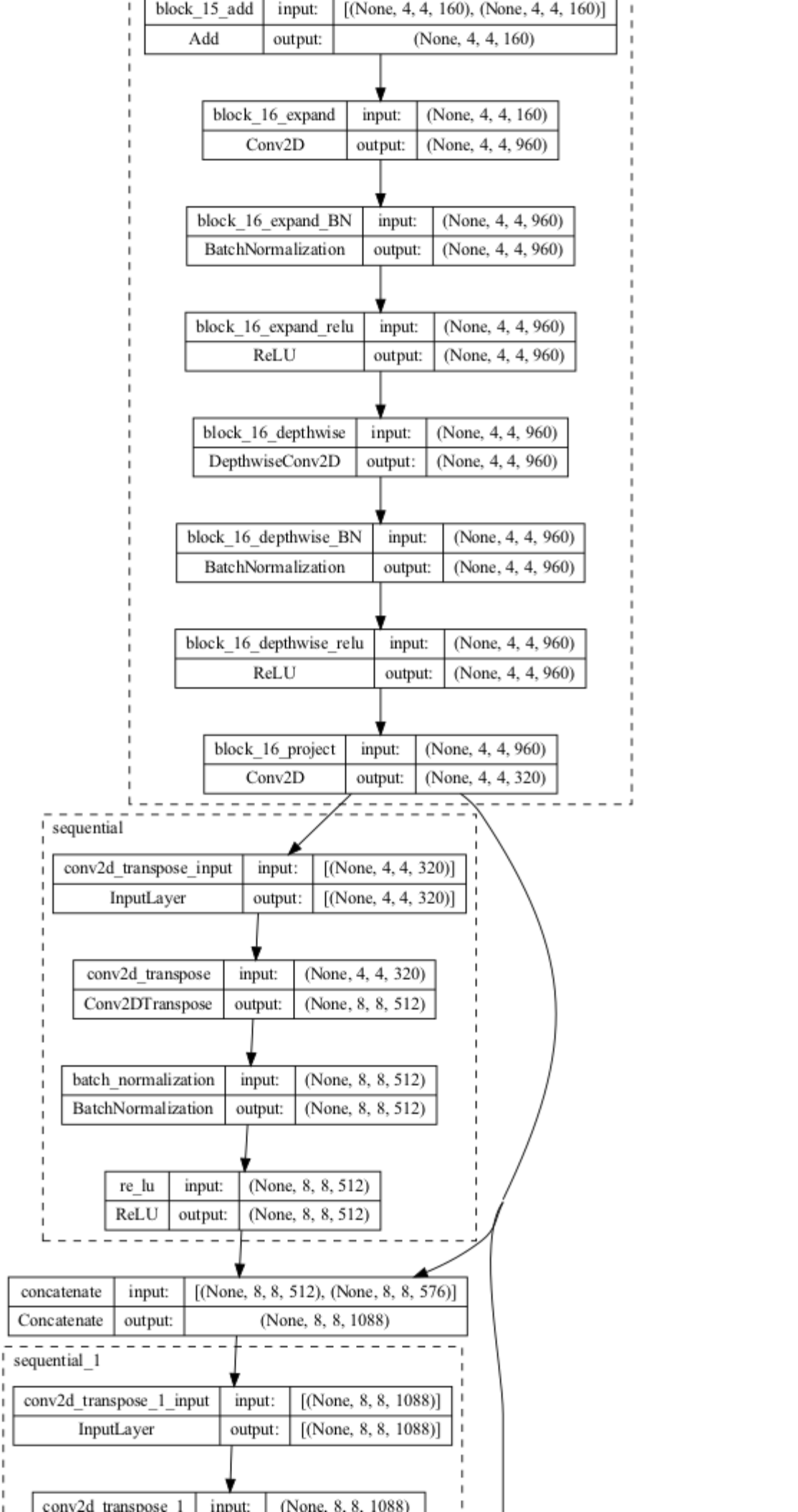

block_7_add | input: | [(None, 8, 8, 64), (None, 8, 8, 64)]
Add | output: | (None, 8, 8, 64)
block_8_expand | input: | (None, 8, 8, 64)
Conv2D | output: | (None, 8, 8, 384)
block_8_expand_BN | input: | (None, 8, 8, 384)
BatchNormalization | output: | (None, 8, 8, 384)
block_8_expand_relu | input: | (None, 8, 8, 384)
ReLU | output: | (None, 8, 8, 384)
block_8_depthwise | input: | (None, 8, 8, 384)
DepthwiseConv2D | output: | (None, 8, 8, 384)
block_8_depthwise_BN | input: | (None, 8, 8, 384)
BatchNormalization | output: | (None, 8, 8, 384)
block_8_depthwise_relu | input: | (None, 8, 8, 384)
ReLU | output: | (None, 8, 8, 384)
block_8_project | input: | (None, 8, 8, 384)
Conv2D | output: | (None, 8, 8, 64)
block_8_project_BN | input: | (None, 8, 8, 64)
BatchNormalization | output: | (None, 8, 8, 64)
block_8_add | input: | [(None, 8, 8, 64), (None, 8, 8, 64)]
Add | output: | (None, 8, 8, 64)
block_9_expand | input: | (None, 8, 8, 64)
Conv2D | output: | (None, 8, 8, 384)
block_9_expand_BN | input: | (None, 8, 8, 384)
BatchNormalization | output: | (None, 8, 8, 384)
block_9_expand_relu | input: | (None, 8, 8, 384)
ReLU | output: | (None, 8, 8, 384)
block_9_depthwise | input: | (None, 8, 8, 384)
DepthwiseConv2D | output: | (None, 8, 8, 384)

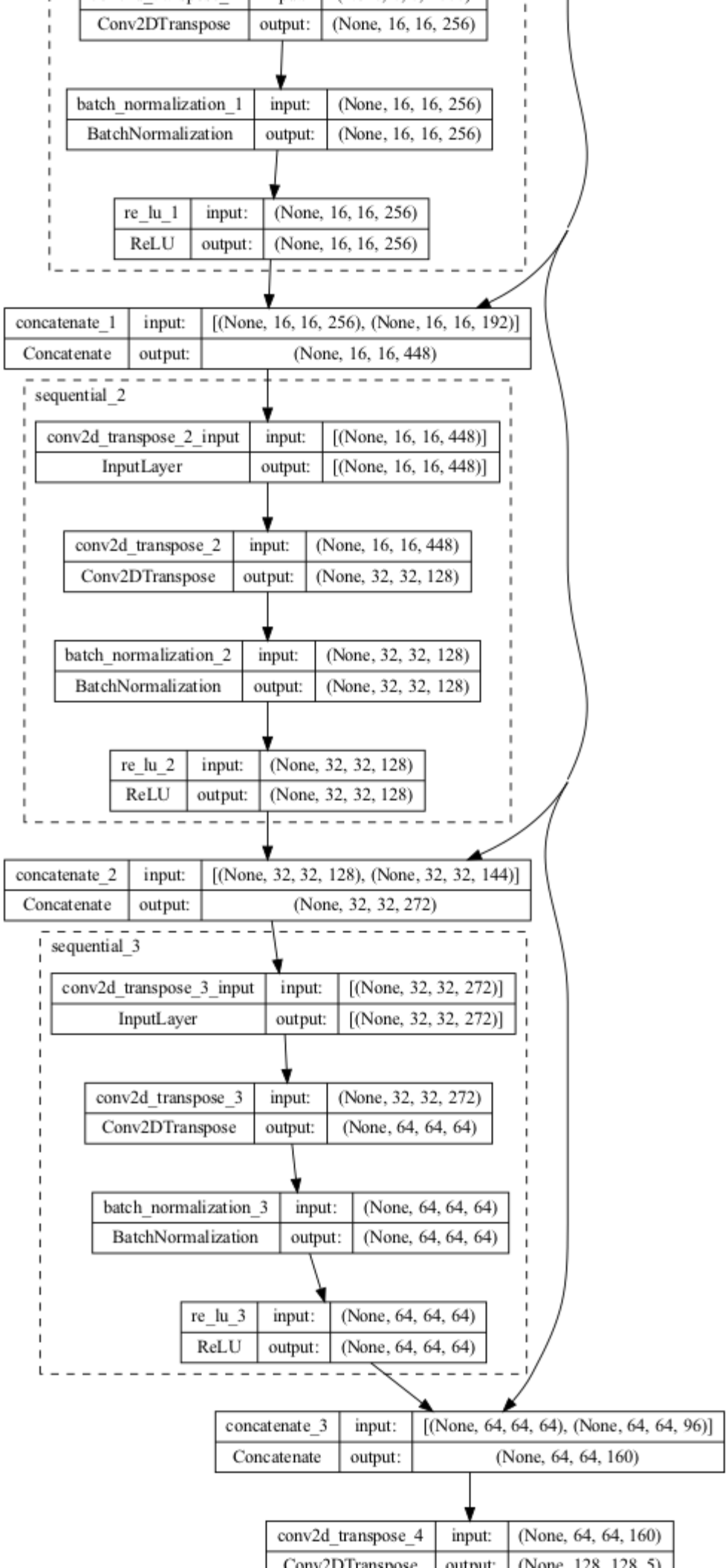

block_9_depthwise_BN | input: | (None, 8, 8, 384)
BatchNormalization | output: | (None, 8, 8, 384)
block_9_depthwise_relu | input: | (None, 8, 8, 384)
ReLU | output: | (None, 8, 8, 384)
block_9_project | input: | (None, 8, 8, 384)
Conv2D | output: | (None, 8, 8, 64)
block_9_project_BN | input: | (None, 8, 8, 64)
BatchNormalization | output: | (None, 8, 8, 64)
block_9_add | input: | [(None, 8, 8, 64), (None, 8, 8, 64)]
Add | output: | (None, 8, 8, 64)
block_10_expand | input: | (None, 8, 8, 64)
Conv2D | output: | (None, 8, 8, 384)
block_10_expand_BN | input: | (None, 8, 8, 384)
BatchNormalization | output: | (None, 8, 8, 384)
block_10_expand_relu | input: | (None, 8, 8, 384)
ReLU | output: | (None, 8, 8, 384)
block_10_depthwise | input: | (None, 8, 8, 384)
DepthwiseConv2D | output: | (None, 8, 8, 384)
block_10_depthwise_BN | input: | (None, 8, 8, 384)
BatchNormalization | output: | (None, 8, 8, 384)
block_10_depthwise_relu | input: | (None, 8, 8, 384)
ReLU | output: | (None, 8, 8, 384)
block_10_project | input: | (None, 8, 8, 384)
Conv2D | output: | (None, 8, 8, 96)
block_10_project_BN | input: | (None, 8, 8, 96)
BatchNormalization | output: | (None, 8, 8, 96)
block_11_expand | input: | (None, 8, 8, 96)
Conv2D | output: | (None, 8, 8, 576)


block_11_expand_BN | input: | (None, 8, 8, 576)
BatchNormalization | output: | (None, 8, 8, 576)
block_11_expand_relu | input: | (None, 8, 8, 576)
ReLU | output: | (None, 8, 8, 576)
block_11_depthwise | input: | (None, 8, 8, 576)
DepthwiseConv2D | output: | (None, 8, 8, 576)
block_11_depthwise_BN | input: | (None, 8, 8, 576)
BatchNormalization | output: | (None, 8, 8, 576)
block_11_depthwise_relu | input: | (None, 8, 8, 576)
ReLU | output: | (None, 8, 8, 576)
block_11_project | input: | (None, 8, 8, 576)
Conv2D | output: | (None, 8, 8, 96)
block_11_project_BN | input: | (None, 8, 8, 96)
BatchNormalization | output: | (None, 8, 8, 96)
block_11_add | input: | [(None, 8, 8, 96), (None, 8, 8, 96)]
Add | output: | (None, 8, 8, 96)
block_12_expand | input: | (None, 8, 8, 96)
Conv2D | output: | (None, 8, 8, 576)
block_12_expand_BN | input: | (None, 8, 8, 576)
BatchNormalization | output: | (None, 8, 8, 576)
block_12_expand_relu | input: | (None, 8, 8, 576)
ReLU | output: | (None, 8, 8, 576)
block_12_depthwise | input: | (None, 8, 8, 576)
DepthwiseConv2D | output: | (None, 8, 8, 576)
block_12_depthwise_BN | input: | (None, 8, 8, 576)
BatchNormalization | output: | (None, 8, 8, 576)
block_12_depthwise_relu | input: | (None, 8, 8, 576)
ReLU | output: | (None, 8, 8, 576)
block_12_project | input: | (None, 8, 8, 576)


block_12_project_BN | input: | (None, 8, 8, 96)
BatchNormalization | output: | (None, 8, 8, 96)
block_12_add | input: | [(None, 8, 8, 96), (None, 8, 8, 96)]
Add | output: | (None, 8, 8, 96)
block_13_expand | input: | (None, 8, 8, 96)
Conv2D | output: | (None, 8, 8, 576)
block_13_expand_BN | input: | (None, 8, 8, 576)
BatchNormalization | output: | (None, 8, 8, 576)
block_13_expand_relu | input: | (None, 8, 8, 576)
ReLU | output: | (None, 8, 8, 576)
block_13_pad | input: | (None, 8, 8, 576)
ZeroPadding2D | output: | (None, 9, 9, 576)
block_13_depthwise | input: | (None, 9, 9, 576)
DepthwiseConv2D | output: | (None, 4, 4, 576)
block_13_depthwise_BN | input: | (None, 4, 4, 576)
BatchNormalization | output: | (None, 4, 4, 576)
block_13_depthwise_relu | input: | (None, 4, 4, 576)
ReLU | output: | (None, 4, 4, 576)
block_13_project | input: | (None, 4, 4, 576)
Conv2D | output: | (None, 4, 4, 160)
block_13_project_BN | input: | (None, 4, 4, 160)
BatchNormalization | output: | (None, 4, 4, 160)
block_14_expand | input: | (None, 4, 4, 160)
Conv2D | output: | (None, 4, 4, 960)
block_14_expand_BN | input: | (None, 4, 4, 960)
BatchNormalization | output: | (None, 4, 4, 960)
block_14_expand_relu | input: | (None, 4, 4, 960)
ReLU | output: | (None, 4, 4, 960)


block_14_depthwise | input: | (None, 4, 4, 960)
DepthwiseConv2D | output: | (None, 4, 4, 960)
block_14_depthwise_BN | input: | (None, 4, 4, 960)
BatchNormalization | output: | (None, 4, 4, 960)
block_14_depthwise_relu | input: | (None, 4, 4, 960)
ReLU | output: | (None, 4, 4, 960)
block_14_project | input: | (None, 4, 4, 960)
Conv2D | output: | (None, 4, 4, 160)
block_14_project_BN | input: | (None, 4, 4, 160)
BatchNormalization | output: | (None, 4, 4, 160)
block_14_add | input: | [(None, 4, 4, 160), (None, 4, 4, 160)]
Add | output: | (None, 4, 4, 160)
block_15_expand | input: | (None, 4, 4, 160)
Conv2D | output: | (None, 4, 4, 960)
block_15_expand_BN | input: | (None, 4, 4, 960)
BatchNormalization | output: | (None, 4, 4, 960)
block_15_expand_relu | input: | (None, 4, 4, 960)
ReLU | output: | (None, 4, 4, 960)
block_15_depthwise | input: | (None, 4, 4, 960)
DepthwiseConv2D | output: | (None, 4, 4, 960)
block_15_depthwise_BN | input: | (None, 4, 4, 960)
BatchNormalization | output: | (None, 4, 4, 960)
block_15_depthwise_relu | input: | (None, 4, 4, 960)
ReLU | output: | (None, 4, 4, 960)
block_15_project | input: | (None, 4, 4, 960)
Conv2D | output: | (None, 4, 4, 160)
block_15_project_BN | input: | (None, 4, 4, 160)
BatchNormalization | output: | (None, 4, 4, 160)


block_15_add | input: | [(None, 4, 4, 160), (None, 4, 4, 160)]
Add | output: | (None, 4, 4, 160)
block_16_expand | input: | (None, 4, 4, 160)
Conv2D | output: | (None, 4, 4, 960)
block_16_expand_BN | input: | (None, 4, 4, 960)
BatchNormalization | output: | (None, 4, 4, 960)
block_16_expand_relu | input: | (None, 4, 4, 960)
ReLU | output: | (None, 4, 4, 960)
block_16_depthwise | input: | (None, 4, 4, 960)
DepthwiseConv2D | output: | (None, 4, 4, 960)
block_16_depthwise_BN | input: | (None, 4, 4, 960)
BatchNormalization | output: | (None, 4, 4, 960)
block_16_depthwise_relu | input: | (None, 4, 4, 960)
ReLU | output: | (None, 4, 4, 960)
block_16_project | input: | (None, 4, 4, 960)
Conv2D | output: | (None, 4, 4, 320)
sequential
conv2d_transpose_input | input: | [(None, 4, 4, 320)]
InputLayer | output: | [(None, 4, 4, 320)]
conv2d_transpose | input: | (None, 4, 4, 320)
Conv2DTranspose | output: | (None, 8, 8, 512)
batch_normalization | input: | (None, 8, 8, 512)
BatchNormalization | output: | (None, 8, 8, 512)
re_lu | input: | (None, 8, 8, 512)
ReLU | output: | (None, 8, 8, 512)
concatenate | input: | [(None, 8, 8, 512), (None, 8, 8, 576)]
Concatenate | output: | (None, 8, 8, 1088)
sequential_1
conv2d_transpose_1_input | input: | [(None, 8, 8, 1088)]
InputLayer | output: | [(None, 8, 8, 1088)]
conv2d_transpose_1 | input: | (None, 8, 8, 1088)


Conv2DTranspose output: (None, 16, 16, 256)
batch_normalization_1 input: (None, 16, 16, 256)
BatchNormalization output: (None, 16, 16, 256)
re_lu_1 input: (None, 16, 16, 256)
ReLU output: (None, 16, 16, 256)
concatenate_1 input: [(None, 16, 16, 256), (None, 16, 16, 192)]
Concatenate output: (None, 16, 16, 448)
sequential_2
conv2d_transpose_2_input input: [(None, 16, 16, 448)]
InputLayer output: [(None, 16, 16, 448)]
conv2d_transpose_2 input: (None, 16, 16, 448)
Conv2DTranspose output: (None, 32, 32, 128)
batch_normalization_2 input: (None, 32, 32, 128)
BatchNormalization output: (None, 32, 32, 128)
re_lu_2 input: (None, 32, 32, 128)
ReLU output: (None, 32, 32, 128)
concatenate_2 input: [(None, 32, 32, 128), (None, 32, 32, 144)]
Concatenate output: (None, 32, 32, 272)
sequential_3
conv2d_transpose_3_input input: [(None, 32, 32, 272)]
InputLayer output: [(None, 32, 32, 272)]
conv2d_transpose_3 input: (None, 32, 32, 272)
Conv2DTranspose output: (None, 64, 64, 64)
batch_normalization_3 input: (None, 64, 64, 64)
BatchNormalization output: (None, 64, 64, 64)
re_lu_3 input: (None, 64, 64, 64)
ReLU output: (None, 64, 64, 64)
concatenate_3 input: [(None, 64, 64, 64), (None, 64, 64, 96)]
Concatenate output: (None, 64, 64, 160)
conv2d_transpose_4 input: (None, 64, 64, 160)

Conv2D Transpose	Output:	(None, 128, 128, 3)

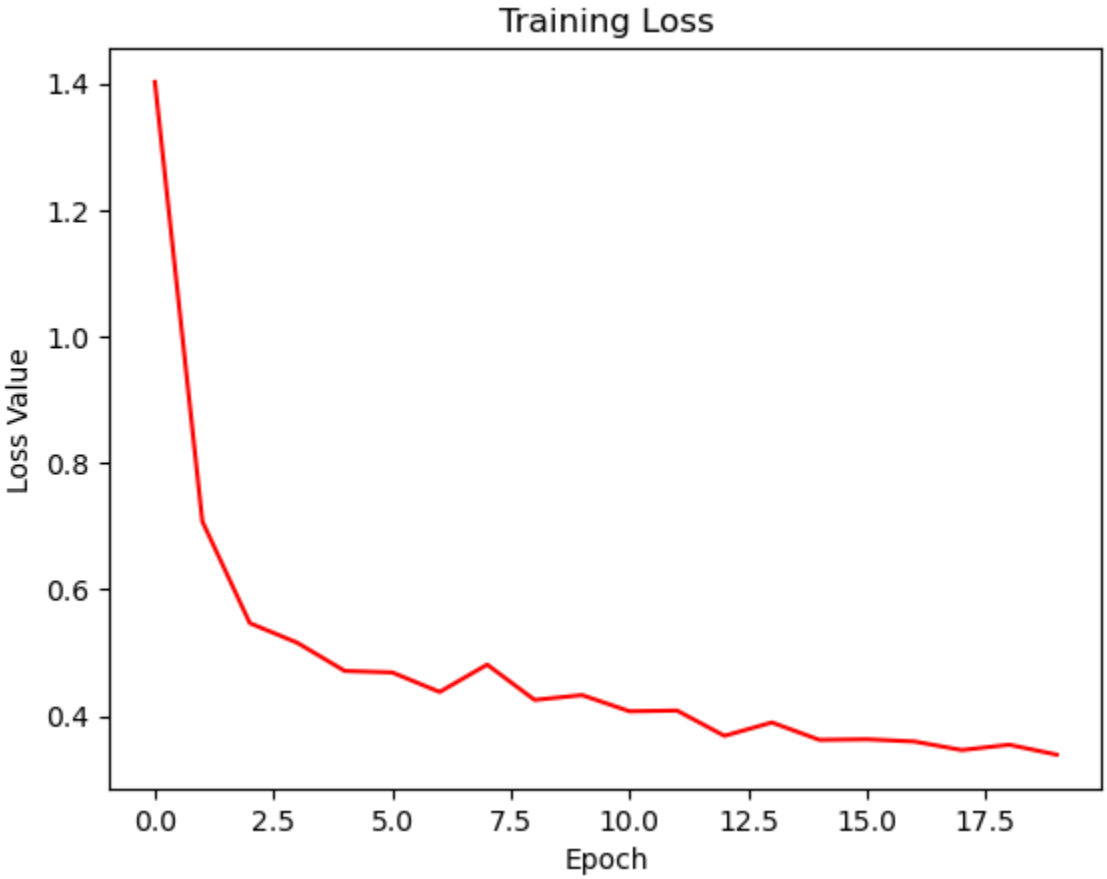
In [14]:
```
# Train the model.
model_history = model.fit(
    train_ds,
    validation_data=test_ds,
    epochs=EPOCHS,
    steps_per_epoch=steps_per_epoch,
    validation_steps=validation_steps
)
```

```
Epoch 1/20
2025-03-31 05:48:23.931704: W tensorflow/tsl/platform/profile_utils/cpu_utils.cc:128] Fai
led to get CPU frequency: 0 Hz
4/4 [==============================] - 4s 688ms/step - loss: 1.4026 - accuracy: 0.3716
Epoch 2/20
4/4 [==============================] - 3s 797ms/step - loss: 0.7076 - accuracy: 0.8350
Epoch 3/20
4/4 [==============================] - 3s 773ms/step - loss: 0.5467 - accuracy: 0.8859
Epoch 4/20
4/4 [==============================] - 3s 720ms/step - loss: 0.5157 - accuracy: 0.8843
Epoch 5/20
4/4 [==============================] - 2s 603ms/step - loss: 0.4711 - accuracy: 0.8896
Epoch 6/20
4/4 [==============================] - 3s 760ms/step - loss: 0.4687 - accuracy: 0.8865
Epoch 7/20
4/4 [==============================] - 3s 725ms/step - loss: 0.4378 - accuracy: 0.8926
Epoch 8/20
4/4 [==============================] - 3s 683ms/step - loss: 0.4811 - accuracy: 0.8772
Epoch 9/20
4/4 [==============================] - 3s 651ms/step - loss: 0.4253 - accuracy: 0.8921
Epoch 10/20
4/4 [==============================] - 3s 615ms/step - loss: 0.4329 - accuracy: 0.8864
Epoch 11/20
4/4 [==============================] - 3s 758ms/step - loss: 0.4073 - accuracy: 0.8887
Epoch 12/20
4/4 [==============================] - 2s 656ms/step - loss: 0.4084 - accuracy: 0.8824
Epoch 13/20
4/4 [==============================] - 3s 761ms/step - loss: 0.3684 - accuracy: 0.8930
Epoch 14/20
4/4 [==============================] - 3s 650ms/step - loss: 0.3896 - accuracy: 0.8815
Epoch 15/20
4/4 [==============================] - 3s 626ms/step - loss: 0.3621 - accuracy: 0.8890
Epoch 16/20
4/4 [==============================] - 3s 749ms/step - loss: 0.3630 - accuracy: 0.8863
Epoch 17/20
4/4 [==============================] - 3s 799ms/step - loss: 0.3595 - accuracy: 0.8866
Epoch 18/20
4/4 [==============================] - 3s 698ms/step - loss: 0.3459 - accuracy: 0.8890
Epoch 19/20
4/4 [==============================] - 3s 663ms/step - loss: 0.3544 - accuracy: 0.8876
Epoch 20/20
4/4 [==============================] - 3s 659ms/step - loss: 0.3387 - accuracy: 0.8905
```

4.0 Evaluate Model

4.1 Plot model loss.

In [15]:
```
# Code taken from reference 27.
# Get the loss history.
loss = model_history.history['loss']

# Plot the loss history.
plt.figure()
plt.plot(model_history.epoch, loss, 'r', label='Training loss')
plt.title('Training Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss Value')
plt.show()
```

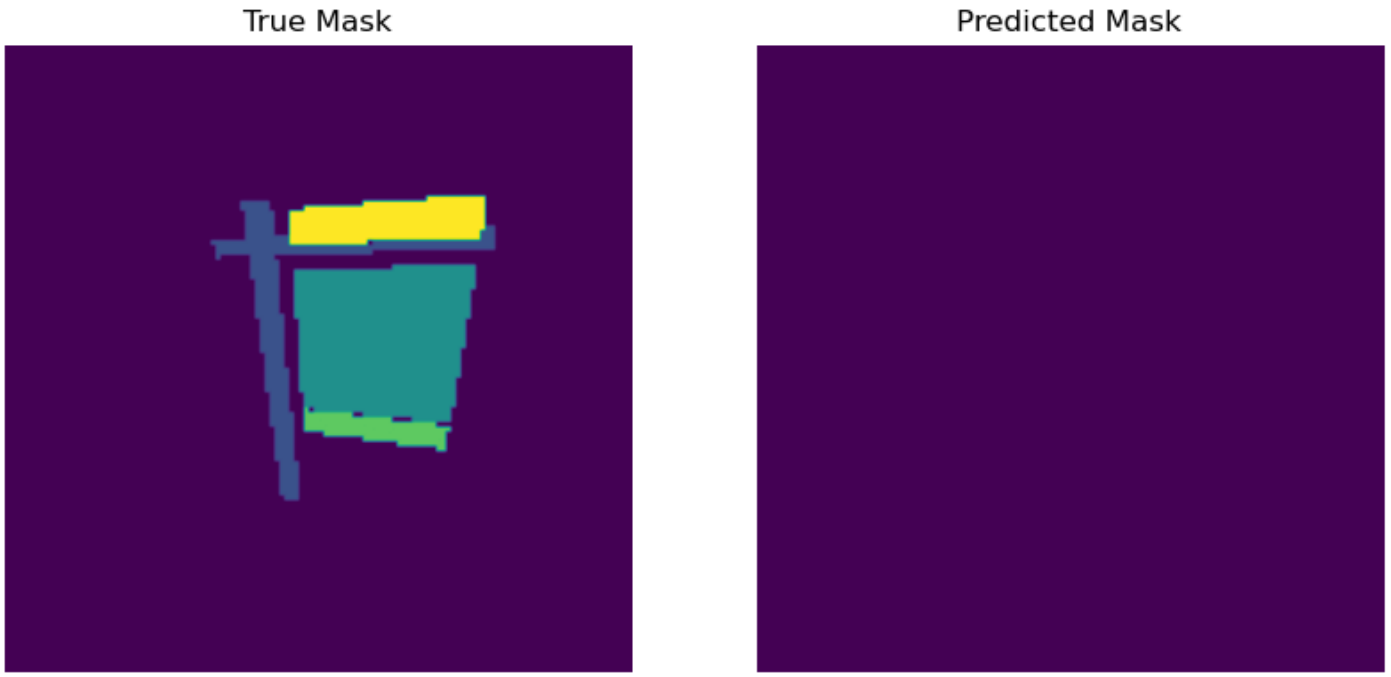


4.2 Plot metrics per class

In [23]:
```
# Guidance for this code provided by reference 29.
# Flatten masks and predictions
y_true_all = []
y_predictions_all = []

# Loop through the test data and predict the masks with the model.
for images, masks in test_ds.take(1):
    predictions = model.predict(images)
    predictions = tf.argmax(predictions, axis=-1).numpy()
    masks = masks.numpy()
    y_true_all.append(masks)
    y_predictions_all.append(predictions)

    # Plot a predicted mask vs an accurate mask

    # Create plot.
    fig, axes = plt.subplots(1, 2, figsize = (10, 5))
```

```
    # Show base image
    axes[0].imshow(masks[4])
    axes[0].set_title("True Mask")
    axes[0].axis('off')

    # Show image with mask.
    axes[1].imshow(predictions[4])
    axes[1].set_title("Predicted Mask")
    axes[1].axis('off')

    plt.show()

# Flatten all masks and predictions
y_true_all = np.concatenate(y_true_all).flatten()
y_predictions_all = np.concatenate(y_predictions_all).flatten()

# Per-Class Accuracy + Precision/Recall
print("Per-Class Metrics:")
print(classification_report(y_true_all, y_predictions_all))
```

```
1/1 [==============================] - 0s 251ms/step
```



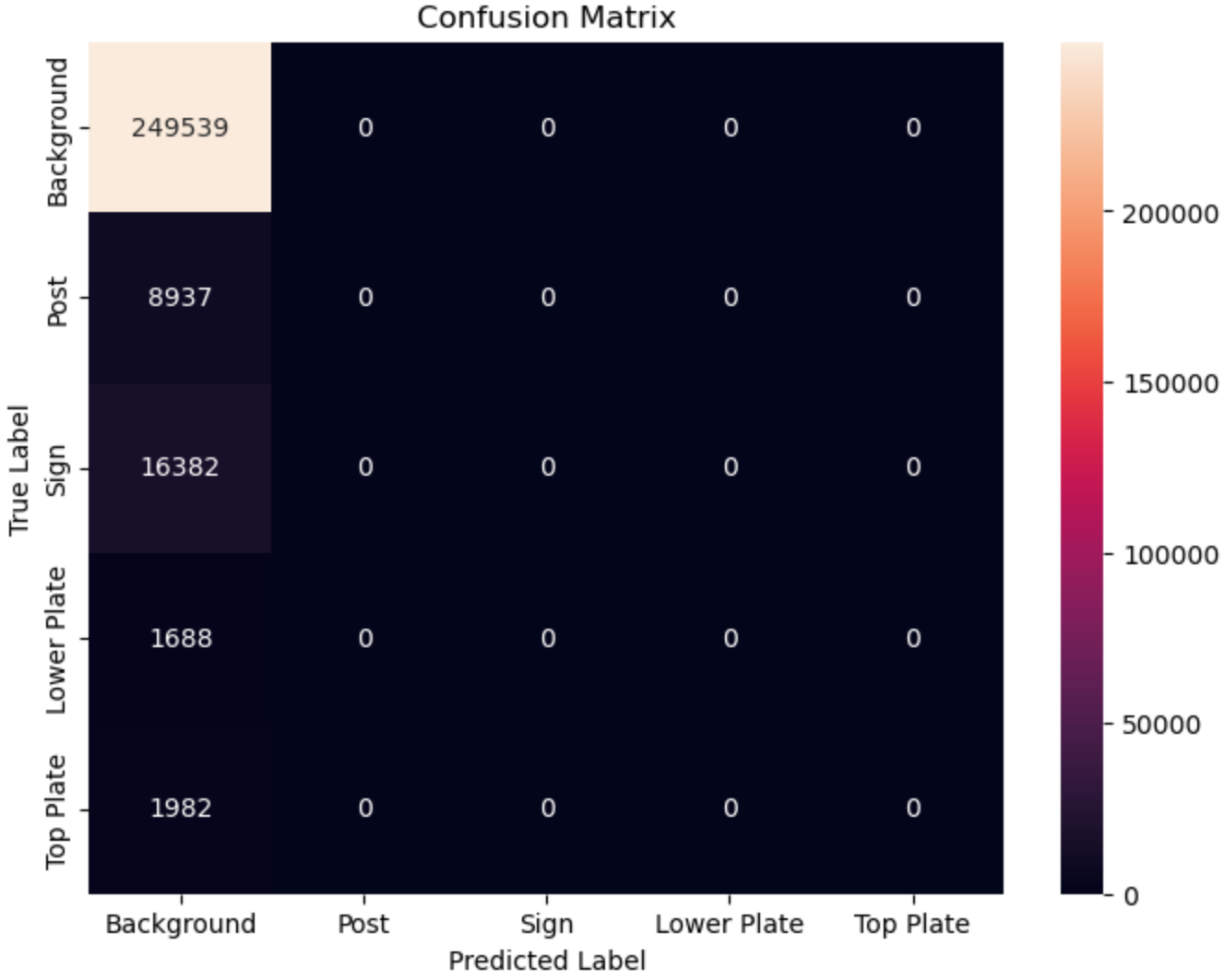

```
Per-Class Metrics:
              precision    recall  f1-score   support

           0       0.90      1.00      0.95    249539
           1       0.00      0.00      0.00      8937
           2       0.00      0.00      0.00     16382
           3       0.00      0.00      0.00      1688
           4       0.00      0.00      0.00      1982

    accuracy                           0.90    278528
   macro avg       0.18      0.20      0.19    278528
weighted avg       0.80      0.90      0.85    278528
```

```
/opt/anaconda3/envs/london/lib/python3.11/site-packages/sklearn/metrics/_classification.p
y:1509: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels w
ith no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/opt/anaconda3/envs/london/lib/python3.11/site-packages/sklearn/metrics/_classification.p
y:1509: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels w
ith no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/opt/anaconda3/envs/london/lib/python3.11/site-packages/sklearn/metrics/_classification.p
y:1509: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels w
ith no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```

4.3 Plot the confusion matrix

In [21]:

```
# Code from reference 28.
# Compute confusion matrix
cm = confusion_matrix(y_true_all, y_predictions_all)

# Define class labels.
class_names = ['Background', 'Post', 'Sign', 'Lower Plate', 'Top Plate']

# Plot the confusion matrix.
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', xticklabels=class_names, yticklabels=class_names)
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.title('Confusion Matrix')
plt.show()
```


Confusion Matrix
Background
Post
Sign
Lower Plate
Top Plate
True Label
249539
0
0
0
0
8937
0
0
0
0
16382
0
0
0
0
1688
0
0
0
0
1982
0
0
0
0
Background
Post
Sign
Lower Plate
Top Plate
Predicted Label
200000
150000
100000
50000
0