

CA3

November 5, 2024

1 Compulsory Assignment 3: Semantic segmentation

Please fill out the the group name, number, members and optionally the name below.

Group number: 30

Group member 1: Peder Ørmen Bukaasen **Group member 2:** Bård Tollef Pedersen

Group member 3: Evind Lid Trøen

Group name (optional): Pesi Morti

2 Assignment Submission

To complete this assignment answer the relevant questions in this notebook and write the code required to implement the relevant models. This is the biggest assignment of the semester, and therefore you get two weeks to work on it. However, we recommend that **you start early**. This assignment has three semi-big sections, each of which build upon the last. So if you delay the assignment until the day before submission, you will most likely fail. This assignment is completed by doing the following. * Submit notebook as an .ipynb file to canvas. * Submit notebook as an .pdf file to canvas. * Submit the python script you run on ORION to canvas. * Submit the SLURM script you run on ORION to canvas. * Submit at least one of your model predictions to the Kaggle leaderboard, and attain a score that is higher than the *BEAT ME* score.

NOTE: Remember to go through the rules given in the lecture “Introduction to compulsory assignments”, as there are many do’s and don’t’s with regard to how you should present the work you are going to submit.

3 Introduction

In the context of transportation, pilots need to identify possible urgent landing areas in forests. Therefore, scientists have provided us with aerial photos of the forest and accompanying image masks that indicate the presence of birch trees in the images. These masks are binary, with a value of 0 representing areas without birch trees and a value of 1 indicating the presence of birch trees. Our objective is to utilize this dataset to train a model capable of performing semantic segmentation, accurately identifying and delineating birch trees in the images.

3.1 Dataset

In this assignment you will be given 3200 annotated images. The image, and mask dimensions are 128x128 pixels. With each image there follows an annotation mask where each pixel is classified

as 1 (birch tree occurrence) or 0 (not birch tree) The test-dataset contains 800 images, where no ground truth masks are given. To evaluate your model on the test dataset, submit your predictions to the Kaggle leaderboard.

3.2 Assignment tasks

1. Familiarising: Before any modelling, visualise examples of the raw data and masks in your Jupyter notebook.
2. Basic U-NET: Create a U-Net with optional dropout. Use a proportion of the images for validation when training. The minimum requirement is to tune the following parameters: number of convolutional filters, dropout and learning rate. Report strategies and scores leading up to the final choice.
3. I recommend adding augmentation (both images and masks) and changing the loss function, e.g. implementing F1 loss. Create a plot showing your model's predicted mask on some images in the training set and compare to the correct mask.
4. Transfer learning: Create a U-Net where the encoder part of the U-Net uses a pretrained VGG16 (or some other similar pre-trained model)
5. This can be achieved by loading a pre-trained model, dropping the final layer and freezing the weights. Add skip connections from the Conv2D layers of the VGG16- based encoder to the expansion layers of the decoder.

3.3 Submissions to the Kaggle leaderboard

Link to the Kaggle leaderboard will be posted in the Canvas assignment.

```
y_pred      = model.predict(X_test)           # Make prediction
flat_y_pred = y_pred.flatten()                # Flatten prediction
flat_y_pred[flat_y_pred >= USER_DETERMINED_THRESHOLD] = 1 # Binarize prediction (Optional, dep
flat_y_pred[flat_y_pred != 1] = 0              # Binarize prediction (Optional, dep
submissionDF = pd.DataFrame()
submissionDF['ID'] = range(len(flat_y_pred))    # The submission csv file must have
submissionDF['Prediction'] = flat_y_pred
submissionDF.to_csv('submission.csv', index=False) # Remember to store the dataframe to
```

4 Library imports

Feel free to import other packages as needed.

```
[1]: import time
from tqdm import tqdm # Cool progress bar
import random
import numpy as np
import pandas as pd
import h5py
import cv2
import re
from IPython.display import Image
import matplotlib.pyplot as plt
```

```

import seaborn as sns
from utilities import *
import tensorflow as tf
import tensorflow.keras as ks
from tensorflow.keras.metrics import FalseNegatives, FalsePositives, TrueNegatives, TruePositives
from tensorflow.keras.applications import EfficientNetB7
from sklearn.utils import shuffle
SEED = 458 # Feel free to set another seed if you want to
RNG = np.random.default_rng(SEED) # Random number generator
tf.random.set_seed(SEED)

```

5 Data loading

Load the data from the HDF5 file `tree_train.h5` and `tree_test.h5` that is available on Canvas, and Kaggle. The data should be loaded in the same manner as in CA2.

```

[2]: FILE_PATH = "tree_train.h5" # If data is in same directory as Jupyter File

with h5py.File(FILE_PATH, 'r') as f:
    print('Datasets in file:', list(f.keys()))
    X_train = np.asarray(f['X'])
    y_train = np.asarray(f['y'])
    print('Nr. train images: %i'%(X_train.shape[0]))

```

Datasets in file: ['X', 'y']
 Nr. train images: 3200

6 Visualization

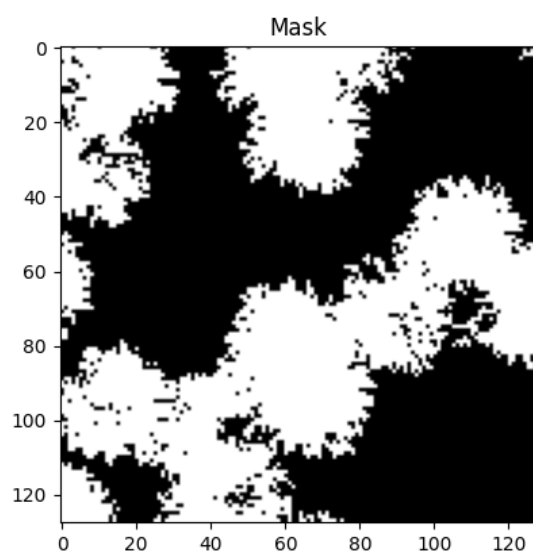
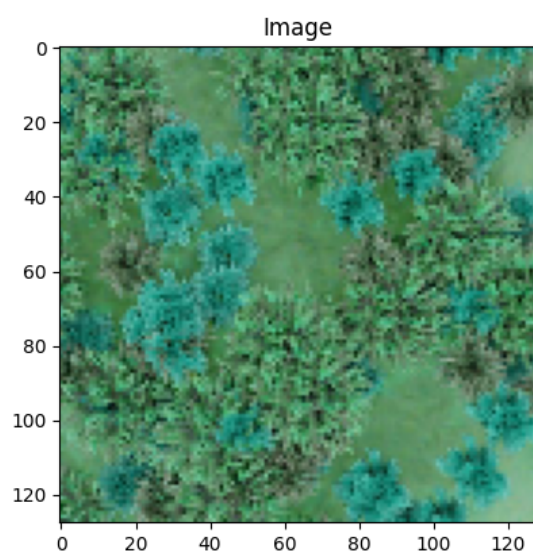
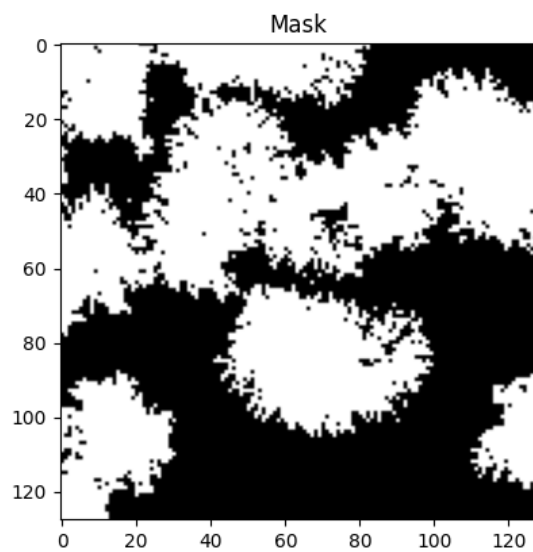
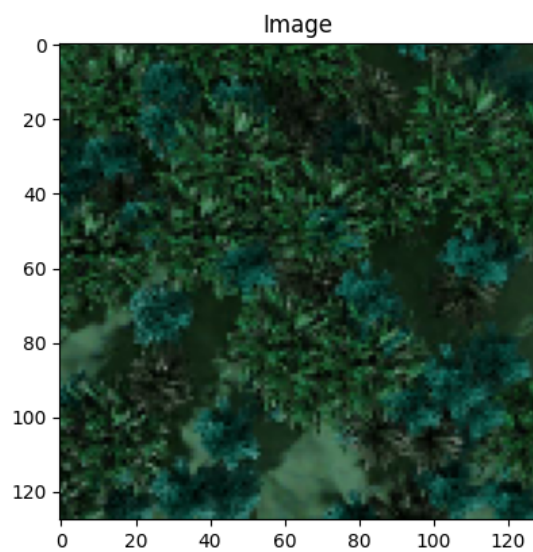
Plot a few samples images and masks.

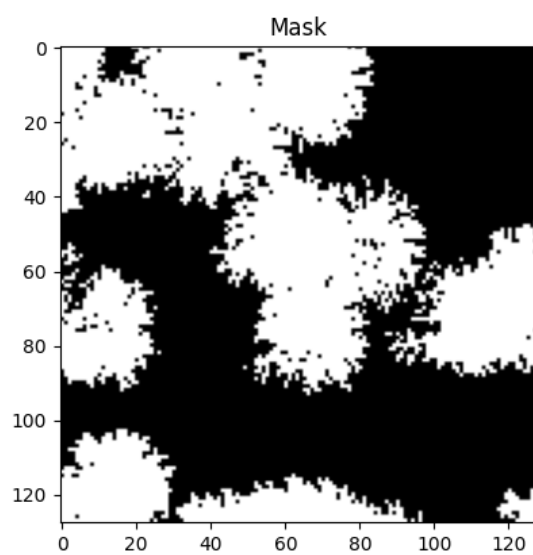
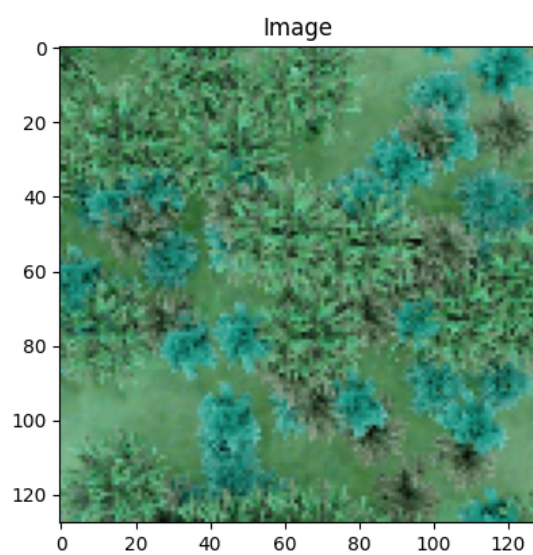
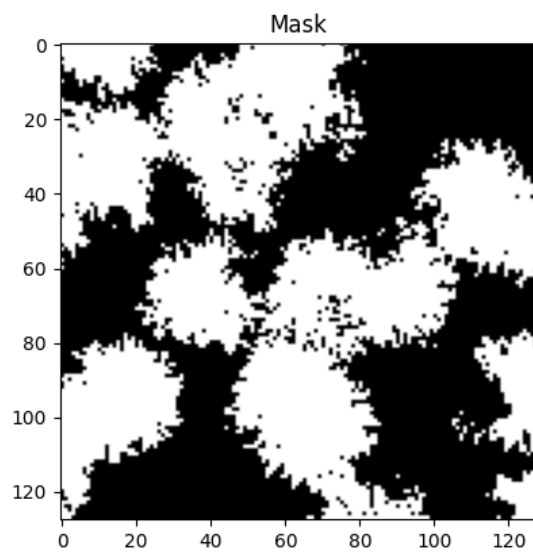
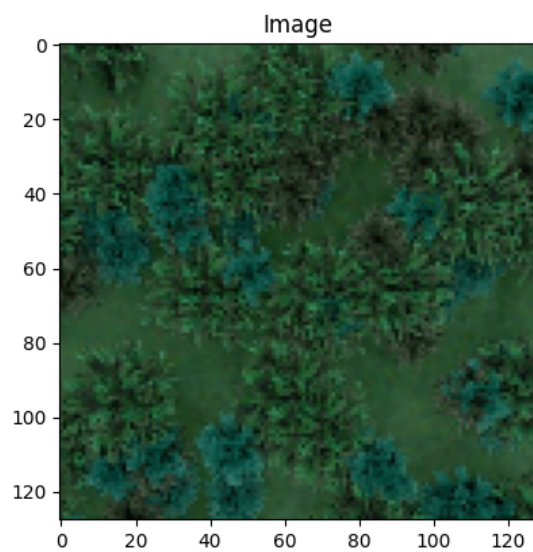
Feel free to visualize any other aspects of the dataset that you feel are relevant.

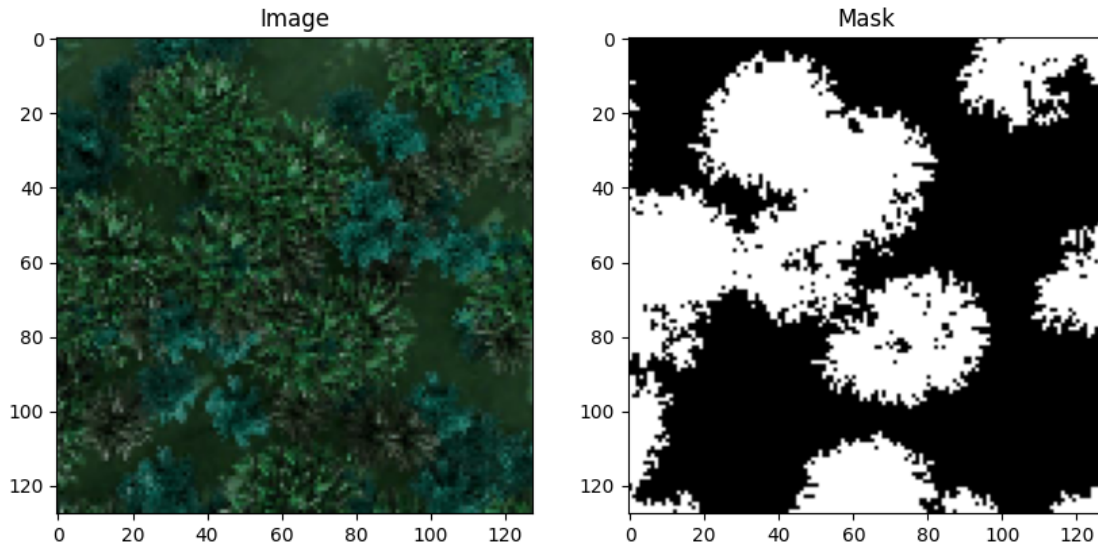
```

[3]: random_numbers = RNG.integers(0, X_train.shape[0], size=5)
for i in random_numbers:
    image = X_train[i]
    mask = y_train[i]
    fig, axes = plt.subplots(1, 2, figsize=(10, 5))
    axes[0].imshow(image)
    axes[0].set_title('Image')
    axes[1].imshow(mask, cmap='gray')
    axes[1].set_title('Mask')
    plt.show()

```







7 Preprocessing

Preprocess the dataset in whatever ways you think are helpful.

Tips: Perhaps you preprocess the different models in different ways?

```
[4]: # Turn to grey scale images
#X_train = np.expand_dims(X_train, -1)
X_train = X_train.astype("float32")/255

# Converting targets from numbers to categorical format
y_train = ks.utils.to_categorical(y_train, len(np.unique(y_train)))

[5]: zoom = tf.keras.layers.RandomZoom(0.5)
flip = tf.keras.layers.RandomFlip("horizontal_and_vertical")
rotate = tf.keras.layers.RandomRotation(1)
translation = tf.keras.layers.RandomTranslation(height_factor=0.2,
↪width_factor=0.2)
contrast = tf.keras.layers.RandomContrast(0.9)

def augment(combined_image):
    augmentations = [zoom, flip, rotate, translation, contrast]
    num_augmentations = np.random.choice([1, 2])
    chosen_augmentations = np.random.choice(augmentations, num_augmentations,
↪replace=False)
    for augmentation in chosen_augmentations:
        combined_image = augmentation(combined_image)

    rotated_x_image = combined_image[..., :3]
```

```

rotated_y_image = combined_image[..., 3:]
return rotated_x_image, rotated_y_image

def agument_all(X_train, y_train, n):
    augmented_X = []
    augmented_y = []

    for i in tqdm(range(n)):
        random_index = RNG.integers(0, X_train.shape[0])
        rotated_x_image, rotated_y_image = augment(np.
concatenate([X_train[random_index], y_train[random_index]], axis=-1))
        augmented_X.append(rotated_x_image)
        augmented_y.append(rotated_y_image)

    # Convert lists to numpy arrays and concatenate with the original arrays
    augmented_X = np.array(augmented_X)
    augmented_y = np.array(augmented_y)

    X_train = np.concatenate([X_train, augmented_X], axis=0)
    y_train = np.concatenate([y_train, augmented_y], axis=0)

    return X_train, y_train

samples_to_generate = 4000
X_train, y_train = agument_all(X_train, y_train, samples_to_generate)

```

100%| | 4000/4000 [00:46<00:00, 85.28it/s]

```

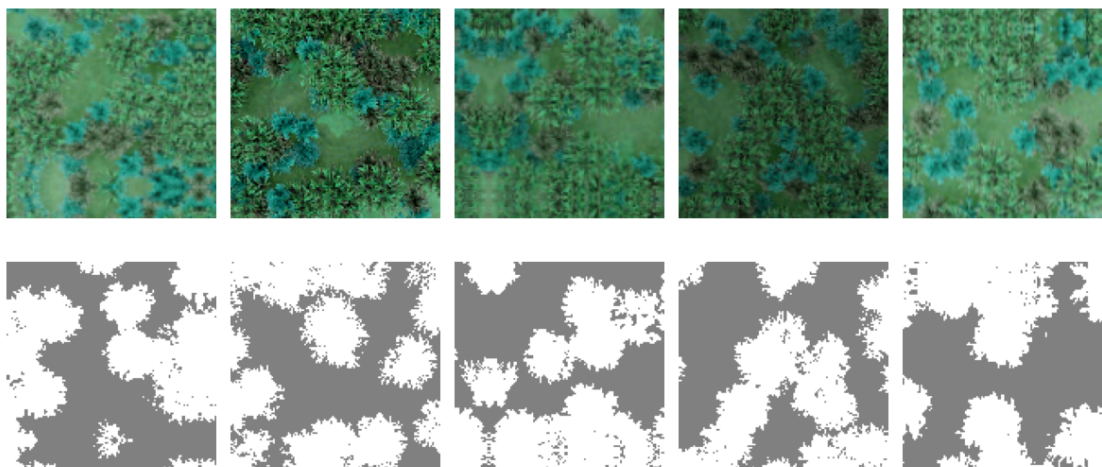
[6]: plt.figure(figsize=(12, 6))

for i in range(5):
    index = - (5 - i)
    plt.subplot(2, 5, i + 1)
    plt.imshow(X_train[index])
    plt.axis("off")

    plt.subplot(2, 5, i + 1 + 5)
    plt.imshow(np.argmax(y_train[index], axis=-1), cmap='gray', alpha=0.5)
    plt.axis("off")

plt.tight_layout()
plt.show()

```

```
[8]: # Shuffle X_train and y_train together
X_train, y_train = shuffle(X_train, y_train, random_state=42)
```

8 Part 1: Implementing U-net

8.1 Intersection over Union

The IoU score is a popular metric in both segmentation and object detection problems.

If you want to use the `plot_training_history` function in the `visualization.py` library remember to compile the model with the TP, TN, FP, FN metrics such that you can estimate the *Intersection-over-Union*. **However, it is voluntary to estimate IoU**

See example below:

```
from tensorflow.keras.metrics import FalseNegatives, FalsePositives, TrueNegatives, TruePositives
from utilities import F1_score,
from visualization import plot_training_history,
...
model.compile(optimizer='Something',
              loss='Something else',
              metrics=[FalseNegatives(),
                      FalsePositives(),
                      TrueNegatives(),
                      TruePositives(),
                      F1_score,
                      OtherMetricOfChoice])

training_history = model.fit(X_train, y_train, ...)
plot_training_history(training_history)
```

You have also been provided with a custom F1-score metric in the `utilities.py` library, which is specific for image segmentation. **This is mandatory to use when compiling the model.**

8.2 Task 1.1 Model implementation

Implement the classical U-net structure that you have learned about in the lectures. Feel free to experiment with the number of layers, loss-function, batch-normalization, etc. **Remember to compile with the F1-score metric.**

8.2.1 U-net

```
[9]: def double_conv_block(x, n_filters, kernel_size=3, padding="same",  
    ↪activation_ = "relu", batchnorm=True):  
    x = ks.layers.Conv2D(filters = n_filters, kernel_size=[kernel_size_,  
    ↪kernel_size_],  
        padding = padding_, kernel_initializer = "he_normal")(x)  
    if batchnorm:  
        x = ks.layers.BatchNormalization()(x)  
  
    x = ks.layers.Activation(activation_)(x)  
    x = ks.layers.Conv2D(filters = n_filters, kernel_size=[kernel_size_,  
    ↪kernel_size_],  
        padding = padding_, kernel_initializer = "he_normal")(x)  
    if batchnorm:  
        x = ks.layers.BatchNormalization()(x)  
  
    x = ks.layers.Activation(activation_)(x)  
    return x  
  
def downsample_block(x, n_filters, kernel_size_dcb=3, dropout=0.1,  
    ↪kernel_size_mp2d=2):  
    f = double_conv_block(x, n_filters, kernel_size_dcb)  
    p = ks.layers.MaxPool2D((kernel_size_mp2d,kernel_size_mp2d))(f)  
    p = ks.layers.Dropout(dropout)(p)  
    return f, p  
  
def upsample_block(x, conv_features, n_filters, kernel_size_c2dt=3, stride_=2,  
    ↪dropout=0.1):  
    x = ks.layers.Conv2DTranspose(n_filters, (kernel_size_c2dt,  
    ↪kernel_size_c2dt),  
        strides=(stride_, stride_), padding="same")(x)  
    x = ks.layers.concatenate([x, conv_features])  
    x = ks.layers.Dropout(dropout)(x)  
    x = double_conv_block(x, n_filters)  
    return x  
  
[10]: def build_unet_model(image_size, n_filters = 16, dropout = 0.1, batchnorm =  
    ↪True, n_classes = 2):  
    # inputs
```

```

inputs = ks.layers.Input(image_size)

# 1 - downsample
c1, p1 = downsample_block(inputs, n_filters)
# 2 - downsample
c2, p2 = downsample_block(p1, n_filters*2)
# 3 - downsample
c3, p3 = downsample_block(p2, n_filters*4)
# 4 - downsample
c4, p4 = downsample_block(p3, n_filters*8)

c5 = double_conv_block(p4, n_filters*16)

# 6 - upsample
u6 = upsample_block(c5, c4, n_filters*8)
# 7 - upsample
u7 = upsample_block(u6, c3, n_filters*4)
# 8 - upsample
u8 = upsample_block(u7, c2, n_filters*2)
# 9 - upsample
u9 = upsample_block(u8, c1, n_filters)

# outputs
outputs = ks.layers.Conv2D(n_classes, (1,1), padding="same", activation =_
↪"softmax")(u9)
UNET_model = ks.Model(inputs, outputs, name="U-Net")

return UNET_model

```

```

[11]: input_img = (128, 128, 3)
print(input_img)
model_UNET = build_UNET_model(input_img)
model_UNET.summary()

```

```

(128, 128, 3)
Model: "U-Net"

```

```

-----
Layer (type)                Output Shape              Param #   Connected to
=====
input_1 (InputLayer)        [(None, 128, 128, 3)]    0         []
conv2d (Conv2D)             (None, 128, 128, 16)     448        input_1[0][0]
batch_normalization (Batch Normalization) (None, 128, 128, 16)     64        conv2d[0][0]

```

```

Normalization)

activation (Activation)      (None, 128, 128, 16)      0
['batch_normalization[0][0]']

conv2d_1 (Conv2D)           (None, 128, 128, 16)      2320
['activation[0][0]']

batch_normalization_1 (Bat  (None, 128, 128, 16)      64
['conv2d_1[0][0]']
chNormalization)

activation_1 (Activation)    (None, 128, 128, 16)      0
['batch_normalization_1[0][0]']

]

max_pooling2d (MaxPooling2  (None, 64, 64, 16)        0
['activation_1[0][0]']
D)

dropout (Dropout)           (None, 64, 64, 16)        0
['max_pooling2d[0][0]']

conv2d_2 (Conv2D)           (None, 64, 64, 32)        4640
['dropout[0][0]']

batch_normalization_2 (Bat  (None, 64, 64, 32)        128
['conv2d_2[0][0]']
chNormalization)

activation_2 (Activation)    (None, 64, 64, 32)        0
['batch_normalization_2[0][0]']

]

conv2d_3 (Conv2D)           (None, 64, 64, 32)        9248
['activation_2[0][0]']

batch_normalization_3 (Bat  (None, 64, 64, 32)        128
['conv2d_3[0][0]']
chNormalization)

activation_3 (Activation)    (None, 64, 64, 32)        0
['batch_normalization_3[0][0]']

]

max_pooling2d_1 (MaxPoolin  (None, 32, 32, 32)        0
['activation_3[0][0]']
g2D)

```

dropout_1 (Dropout)	(None, 32, 32, 32)	0
['max_pooling2d_1[0][0]']		
conv2d_4 (Conv2D)	(None, 32, 32, 64)	18496
['dropout_1[0][0]']		
batch_normalization_4 (Batch Normalization)	(None, 32, 32, 64)	256
['conv2d_4[0][0]']		
activation_4 (Activation)	(None, 32, 32, 64)	0
['batch_normalization_4[0][0]']		
]		
conv2d_5 (Conv2D)	(None, 32, 32, 64)	36928
['activation_4[0][0]']		
batch_normalization_5 (Batch Normalization)	(None, 32, 32, 64)	256
['conv2d_5[0][0]']		
activation_5 (Activation)	(None, 32, 32, 64)	0
['batch_normalization_5[0][0]']		
]		
max_pooling2d_2 (MaxPooling2D)	(None, 16, 16, 64)	0
['activation_5[0][0]']		
dropout_2 (Dropout)	(None, 16, 16, 64)	0
['max_pooling2d_2[0][0]']		
conv2d_6 (Conv2D)	(None, 16, 16, 128)	73856
['dropout_2[0][0]']		
batch_normalization_6 (Batch Normalization)	(None, 16, 16, 128)	512
['conv2d_6[0][0]']		
activation_6 (Activation)	(None, 16, 16, 128)	0
['batch_normalization_6[0][0]']		
]		
conv2d_7 (Conv2D)	(None, 16, 16, 128)	147584
['activation_6[0][0]']		
batch_normalization_7 (Batch Normalization)	(None, 16, 16, 128)	512

```

['conv2d_7[0][0]']
chNormalization)

activation_7 (Activation) (None, 16, 16, 128) 0
['batch_normalization_7[0][0]']

]

max_pooling2d_3 (MaxPoolin (None, 8, 8, 128) 0
['activation_7[0][0]']
g2D)

dropout_3 (Dropout) (None, 8, 8, 128) 0
['max_pooling2d_3[0][0]']

conv2d_8 (Conv2D) (None, 8, 8, 256) 295168
['dropout_3[0][0]']

batch_normalization_8 (Bat (None, 8, 8, 256) 1024
['conv2d_8[0][0]']
chNormalization)

activation_8 (Activation) (None, 8, 8, 256) 0
['batch_normalization_8[0][0]']

]

conv2d_9 (Conv2D) (None, 8, 8, 256) 590080
['activation_8[0][0]']

batch_normalization_9 (Bat (None, 8, 8, 256) 1024
['conv2d_9[0][0]']
chNormalization)

activation_9 (Activation) (None, 8, 8, 256) 0
['batch_normalization_9[0][0]']

]

conv2d_transpose (Conv2DTr (None, 16, 16, 128) 295040
['activation_9[0][0]']
anspose)

concatenate (Concatenate) (None, 16, 16, 256) 0
['conv2d_transpose[0][0]',
'activation_7[0][0]']

dropout_4 (Dropout) (None, 16, 16, 256) 0
['concatenate[0][0]']

conv2d_10 (Conv2D) (None, 16, 16, 128) 295040

```

```

['dropout_4[0][0]']

batch_normalization_10 (Batch Normalization) (None, 16, 16, 128) 512
['conv2d_10[0][0]']
tchNormalization)

activation_10 (Activation) (None, 16, 16, 128) 0
['batch_normalization_10[0][0]']

conv2d_11 (Conv2D) (None, 16, 16, 128) 147584
['activation_10[0][0]']

batch_normalization_11 (Batch Normalization) (None, 16, 16, 128) 512
['conv2d_11[0][0]']
tchNormalization)

activation_11 (Activation) (None, 16, 16, 128) 0
['batch_normalization_11[0][0]']

conv2d_transpose_1 (Conv2D) (None, 32, 32, 64) 73792
['activation_11[0][0]']
Transpose)

concatenate_1 (Concatenate) (None, 32, 32, 128) 0
['conv2d_transpose_1[0][0]',
)
'activation_5[0][0]']

dropout_5 (Dropout) (None, 32, 32, 128) 0
['concatenate_1[0][0]']

conv2d_12 (Conv2D) (None, 32, 32, 64) 73792
['dropout_5[0][0]']

batch_normalization_12 (Batch Normalization) (None, 32, 32, 64) 256
['conv2d_12[0][0]']
tchNormalization)

activation_12 (Activation) (None, 32, 32, 64) 0
['batch_normalization_12[0][0]']

conv2d_13 (Conv2D) (None, 32, 32, 64) 36928
['activation_12[0][0]']

batch_normalization_13 (Batch Normalization) (None, 32, 32, 64) 256

```

```

['conv2d_13[0][0]']
tchNormalization)

activation_13 (Activation) (None, 32, 32, 64) 0
['batch_normalization_13[0][0]

conv2d_transpose_2 (Conv2D (None, 64, 64, 32) 18464
['activation_13[0][0]']
Transpose)

concatenate_2 (Concatenate (None, 64, 64, 64) 0
['conv2d_transpose_2[0][0]',
)
'activation_3[0][0]']

dropout_6 (Dropout) (None, 64, 64, 64) 0
['concatenate_2[0][0]']

conv2d_14 (Conv2D) (None, 64, 64, 32) 18464
['dropout_6[0][0]']

batch_normalization_14 (Ba (None, 64, 64, 32) 128
['conv2d_14[0][0]']
tchNormalization)

activation_14 (Activation) (None, 64, 64, 32) 0
['batch_normalization_14[0][0]

conv2d_15 (Conv2D) (None, 64, 64, 32) 9248
['activation_14[0][0]']

batch_normalization_15 (Ba (None, 64, 64, 32) 128
['conv2d_15[0][0]']
tchNormalization)

activation_15 (Activation) (None, 64, 64, 32) 0
['batch_normalization_15[0][0]

conv2d_transpose_3 (Conv2D (None, 128, 128, 16) 4624
['activation_15[0][0]']
Transpose)

concatenate_3 (Concatenate (None, 128, 128, 32) 0
['conv2d_transpose_3[0][0]',
)

```



```

'activation_1[0][0]']

dropout_7 (Dropout)          (None, 128, 128, 32)      0
['concatenate_3[0][0]']

conv2d_16 (Conv2D)           (None, 128, 128, 16)     4624
['dropout_7[0][0]']

batch_normalization_16 (Ba   (None, 128, 128, 16)     64
['conv2d_16[0][0]']
tchNormalization)

activation_16 (Activation)    (None, 128, 128, 16)     0
['batch_normalization_16[0][0]']

conv2d_17 (Conv2D)           (None, 128, 128, 16)     2320
['activation_16[0][0]']

batch_normalization_17 (Ba   (None, 128, 128, 16)     64
['conv2d_17[0][0]']
tchNormalization)

activation_17 (Activation)    (None, 128, 128, 16)     0
['batch_normalization_17[0][0]']

conv2d_18 (Conv2D)           (None, 128, 128, 2)      34
['activation_17[0][0]']

```

```

=====
=====
Total params: 2164610 (8.26 MB)
Trainable params: 2161666 (8.25 MB)
Non-trainable params: 2944 (11.50 KB)
-----
-----

```

8.3 Task 1.2 Train the model, and plot the training history

Feel free to use the `plot_training_history` function from the provided library `utilities.py`

8.3.1 Training the model

```

[13]: early_stopping = ks.callbacks.EarlyStopping(monitor='val_F1_score',
                                                    patience=5,
                                                    mode='max',
                                                    restore_best_weights=True)

```

```

# Compile the model
model_unet.compile(optimizer=ks.optimizers.Adam(learning_rate=1e-4),
                    loss='categorical_crossentropy',
                    metrics=[FalseNegatives(),
                             FalsePositives(),
                             TrueNegatives(),
                             TruePositives(),
                             F1_score])

# Train the model
model_unet_history = model_unet.fit(X_train, y_train,
                                    epochs=3, #more on orion
                                    batch_size=32,
                                    validation_split=(1/8),
                                    callbacks=[early_stopping])

model_unet.save('model_unet.keras')

```

Epoch 1/3

```

197/197 [=====] - 166s 810ms/step - loss: 0.4264 -
false_negatives_1: 28236096.0000 - false_positives_1: 11798322.0000 -
true_negatives_1: 74983120.0000 - true_positives_1: 91420896.0000 - F1_score:
0.7961 - val_loss: 0.5790 - val_false_negatives_1: 5974917.0000 -
val_false_positives_1: 3735241.0000 - val_true_negatives_1: 8770683.0000 -
val_true_positives_1: 11010359.0000 - val_F1_score: 0.6422

```

Epoch 2/3

```

197/197 [=====] - 153s 775ms/step - loss: 0.3319 -
false_negatives_1: 23590020.0000 - false_positives_1: 7152234.0000 -
true_negatives_1: 79629208.0000 - true_positives_1: 96066936.0000 - F1_score:
0.8478 - val_loss: 0.3552 - val_false_negatives_1: 3379293.0000 -
val_false_positives_1: 1139617.0000 - val_true_negatives_1: 11366307.0000 -
val_true_positives_1: 13605983.0000 - val_F1_score: 0.8282

```

Epoch 3/3

```

197/197 [=====] - 155s 785ms/step - loss: 0.3074 -
false_negatives_1: 22964666.0000 - false_positives_1: 6526885.0000 -
true_negatives_1: 80254568.0000 - true_positives_1: 96692344.0000 - F1_score:
0.8557 - val_loss: 0.2932 - val_false_negatives_1: 3123385.0000 -
val_false_positives_1: 883709.0000 - val_true_negatives_1: 11622215.0000 -
val_true_positives_1: 13861891.0000 - val_F1_score: 0.8527

```

8.3.2 Plotting performance

```

[14]: rawDF = pd.DataFrame(model_unet_history.history)
      plotDF = pd.DataFrame()

      try:

```

```

# Find the number behind the _ in the keys
number = rawDF.columns[1].split('_')[-1]
rawDF = rawDF.rename(columns={'true_positives_' + number :
↪ 'true_positives', 'true_negatives_' + number: 'true_negatives',
↪ 'false_positives_' + number: 'false_positives', 'false_negatives_' + number:
↪ 'false_negatives',
                                'val_true_positives_' + number:
↪ 'val_true_positives', 'val_true_negatives_' + number: 'val_true_negatives',
↪ 'val_false_positives_' + number: 'val_false_positives',
↪ 'val_false_negatives_' + number: 'val_false_negatives'})
except:
    pass

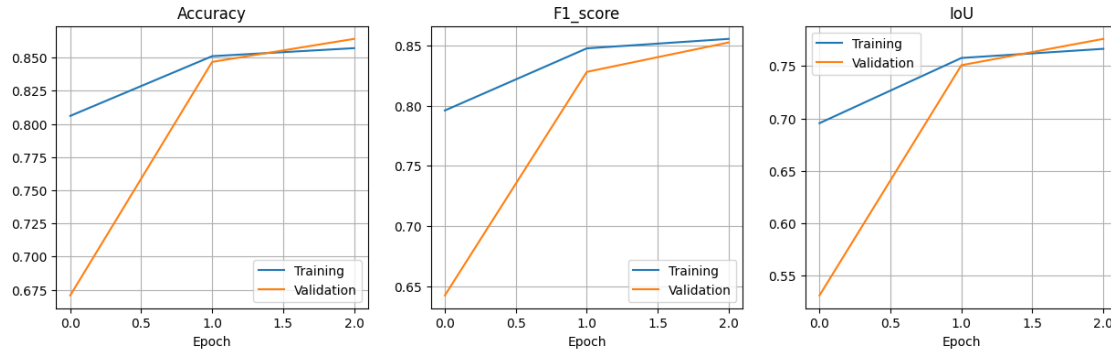
plotDF['Accuracy'] = (rawDF['true_positives'] + rawDF['true_negatives']) /
↪ (rawDF['true_positives'] + rawDF['true_negatives'] +
↪ rawDF['false_positives'] + rawDF['false_negatives'])
plotDF['val_Accuracy'] = (rawDF['val_true_positives'] +
↪ rawDF['val_true_negatives']) / (rawDF['val_true_positives'] +
↪ rawDF['val_true_negatives'] + rawDF['val_false_positives'] +
↪ rawDF['val_false_negatives'])

plotDF['IoU'] = rawDF['true_positives'] / (rawDF['true_positives'] +
↪ rawDF['false_positives'] + rawDF['false_negatives'])
plotDF['val_IoU'] = rawDF['val_true_positives'] /
↪ (rawDF['val_true_positives'] + rawDF['val_false_positives'] +
↪ rawDF['val_false_negatives'])

plotDF['F1_score'] = rawDF['F1_score']
plotDF['val_F1_score'] = rawDF['val_F1_score']

list_of_metrics=['Accuracy', 'F1_score', 'IoU']
train_keys = list_of_metrics
valid_keys = ['val_' + key for key in list_of_metrics]
nr_plots = len(list_of_metrics)
fig, ax = plt.subplots(1,nr_plots,figsize=(5*nr_plots,4))
for i in range(len(list_of_metrics)):
    ax[i].plot(np.array(plotDF[train_keys[i]]), label='Training')
    ax[i].plot(np.array(plotDF[valid_keys[i]]), label='Validation')
    ax[i].set_xlabel('Epoch')
    ax[i].set_title(list_of_metrics[i])
    ax[i].grid('on')
    ax[i].legend()
fig.tight_layout
plt.show()

```



```
[15]: print(model_unet_history.history['F1_score'][-1])
```

```
0.8556792736053467
```

8.4 Task 1.3 Visualize model predictions

Make a plot that illustrates the original image, the predicted mask, and the ground truth mask.

8.4.1 Plotting

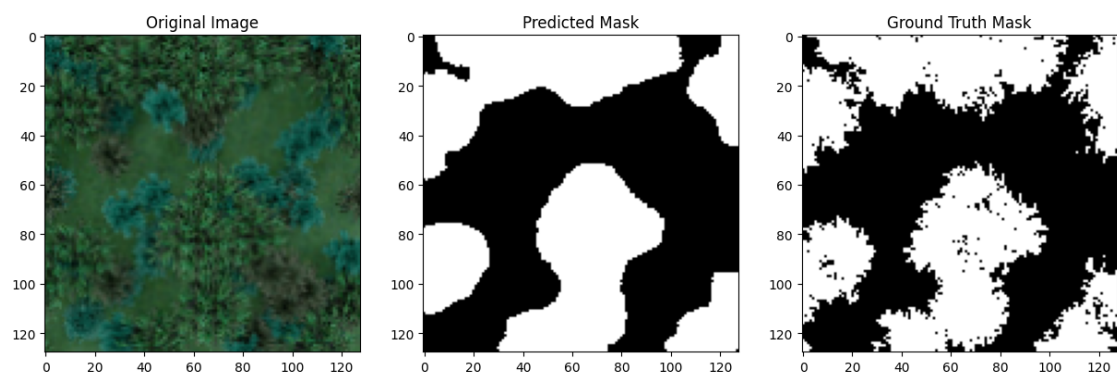
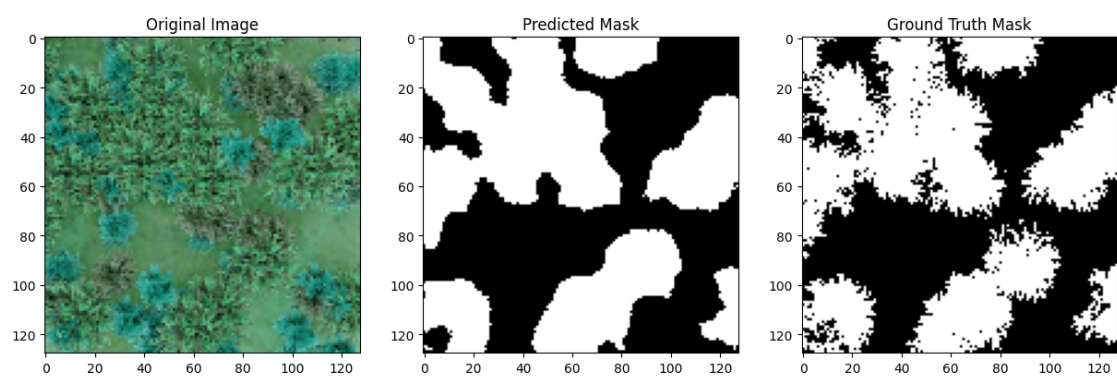
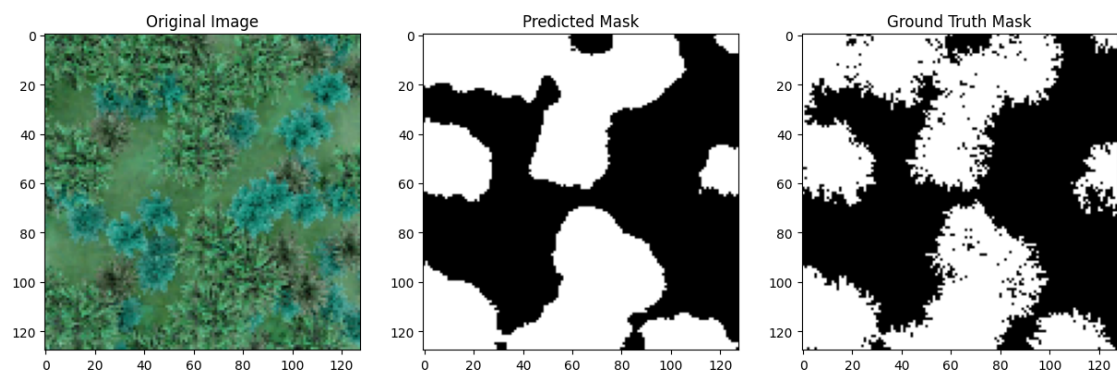
```
[16]: # Predict masks for a few images in the training set
y_pred = model_unet.predict(X_train[:5])

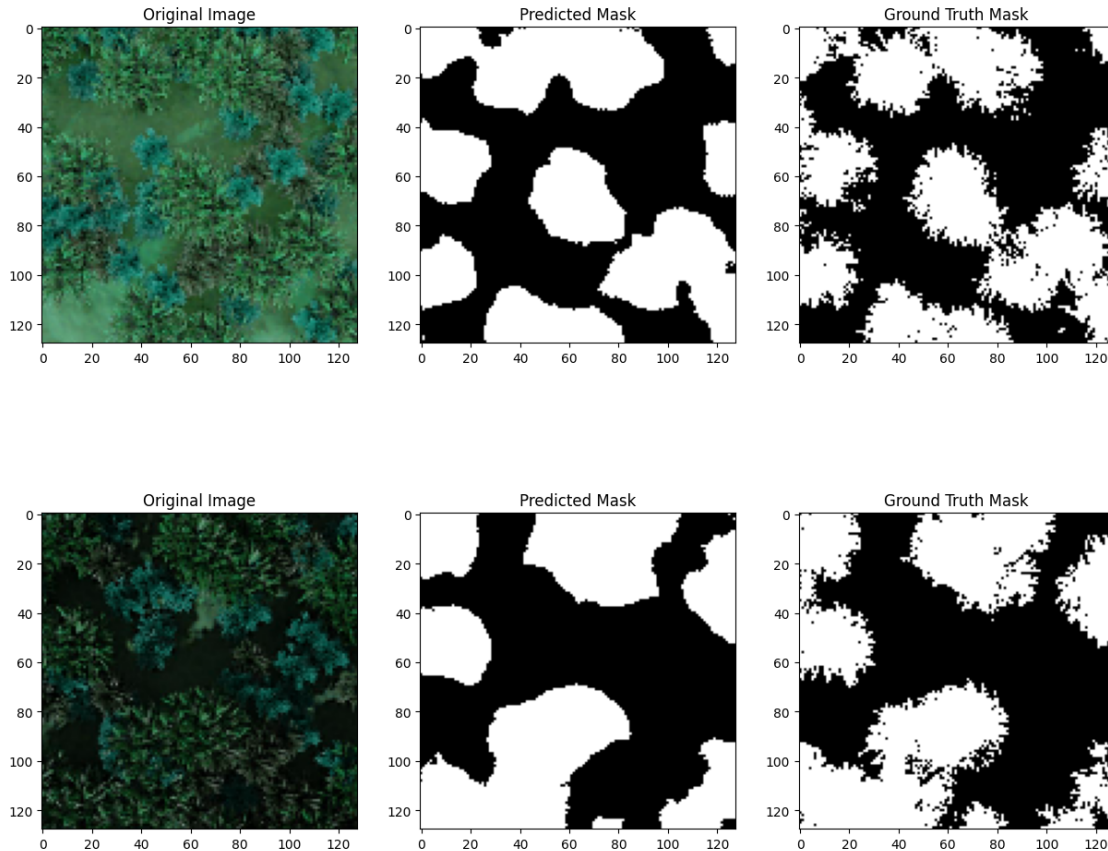
# Convert predicted masks to binary
y_pred_binary = np.argmax(y_pred, axis=-1)

# Convert ground truth masks to binary
y_train_binary = np.argmax(y_train[:5], axis=-1)

# Plot the original image, predicted mask, and ground truth mask for each image
for i in range(5):
    fig, axes = plt.subplots(1, 3, figsize=(15, 5))
    axes[0].imshow(X_train[i].squeeze(), cmap='gray')
    axes[0].set_title('Original Image')
    axes[1].imshow(y_pred_binary[i], cmap='gray')
    axes[1].set_title('Predicted Mask')
    axes[2].imshow(y_train_binary[i], cmap='gray')
    axes[2].set_title('Ground Truth Mask')
    plt.show()
```

```
1/1 [=====] - 2s 2s/step
```





9 Part 2: Implementing U-net with transfer learning

Implement a model with the U-net structure that you have learned about in the lectures, but now with a pre-trained backbone. There are many pre-trained back-bones to choose from. Pick freely from the selection here [tf.keras.applications](https://keras.io/applications/), or here [Keras model scores](#) (nicer table in the second link). Feel free to experiment with the number of layers, loss-function, batch-normalization, etc. Many of the backbones available are quite big, so you might find it quite time-consuming to train them on your personal computers. It might be expedient to only train them for 1-5 epochs on your PCs, and do the full training on Orion in Part 3.

For those with a dedicated graphics card (NVIDIA and AMD) Tensorflow or PyTorch (not syllabus) And wants to experiment with their own compute resources (can be alot of fun)

Tensorflow: <https://learn.microsoft.com/en-us/windows/ai/directml/gpu-tensorflow-plugin>

PyTorch: <https://learn.microsoft.com/en-us/windows/ai/directml/pytorch-windows>

9.1 Task 2.1 Transfer learning model implementation

Implement a U-net model utilizing the pre-trained weights of a publically available network. **Remember to compile with the F1-score metric.**

9.1.1 Transfer learning with U-net

```
[19]: def unet_efficientnetb7(input_shape, dropout_rate=0.1):
    # Input layer
    inputs = ks.layers.Input(shape=input_shape)
    # Pretrained EfficientNetB7 encoder
    base_model = EfficientNetB7(weights='imagenet', include_top=False,
    ↪input_tensor=inputs)
    # Encoder blocks (skip connections)
    block1 = base_model.get_layer('block1a_activation').output # (None, 64,
    ↪64, 64)
    block2 = base_model.get_layer('block2a_activation').output # (None, 32,
    ↪32, 192)
    block3 = base_model.get_layer('block3a_activation').output # (None, 16,
    ↪16, 288)
    block4 = base_model.get_layer('block5a_activation').output # (None, 8, 8,
    ↪960)

    # Middle block
    block5 = base_model.get_layer('top_activation').output # (None, 4, 4,
    ↪256)

    # Decoder blocks
    x5 = upsample_block(block5, block4, 960) # (None, 8, 8, 960)
    x4 = upsample_block(x5, block3, 288) # (None, 16, 16, 288)
    x3 = upsample_block(x4, block2, 192) # (None, 32, 32, 192)
    x2 = upsample_block(x3, block1, 64) # (None, 64, 64, 64)

    # Final upsample and output layer
    x = ks.layers.Conv2DTranspose(32, (3, 3), strides=(2, 2),
    ↪padding="same")(x2) # Final upsample layer (None, 128, 128, 32)
    output = ks.layers.Conv2D(2, (1, 1), padding="same",
    ↪activation="softmax")(x) # Output layer (None, 128, 128, 1)

    # Create the model
    unet_model = ks.models.Model(inputs, output)

    return unet_model

input_img = (128, 128, 3)
model_pre = unet_efficientnetb7(input_img)
model_pre.summary()
```

Model: "model_1"

```
-----
-----
Layer (type)                Output Shape                Param #   Connected to
```


=====			
=====			
input_3 (InputLayer)	[(None, 128, 128, 3)]	0	[]
rescaling_2 (Rescaling)	(None, 128, 128, 3)	0	
['input_3[0][0]']			
normalization_1 (Normaliza	(None, 128, 128, 3)	7	
['rescaling_2[0][0]']			
tion)			
rescaling_3 (Rescaling)	(None, 128, 128, 3)	0	
['normalization_1[0][0]']			
stem_conv_pad (ZeroPadding	(None, 129, 129, 3)	0	
['rescaling_3[0][0]']			
2D)			
stem_conv (Conv2D)	(None, 64, 64, 64)	1728	
['stem_conv_pad[0][0]']			
stem_bn (BatchNormalizatio	(None, 64, 64, 64)	256	
['stem_conv[0][0]']			
n)			
stem_activation (Activatio	(None, 64, 64, 64)	0	
['stem_bn[0][0]']			
n)			
block1a_dwconv (DepthwiseC	(None, 64, 64, 64)	576	
['stem_activation[0][0]']			
onv2D)			
block1a_bn (BatchNormaliza	(None, 64, 64, 64)	256	
['block1a_dwconv[0][0]']			
tion)			
block1a_activation (Activa	(None, 64, 64, 64)	0	
['block1a_bn[0][0]']			
tion)			
block1a_se_squeeze (Global	(None, 64)	0	
['block1a_activation[0][0]']			
AveragePooling2D)			
block1a_se_reshape (Reshap	(None, 1, 1, 64)	0	
['block1a_se_squeeze[0][0]']			
e)			

block1a_se_reduce (Conv2D)	(None, 1, 1, 16)	1040
['block1a_se_reshape[0][0]']		
block1a_se_expand (Conv2D)	(None, 1, 1, 64)	1088
['block1a_se_reduce[0][0]']		
block1a_se_excite (Multipl	(None, 64, 64, 64)	0
['block1a_activation[0][0]',		
y)		
'block1a_se_expand[0][0]']		
block1a_project_conv (Conv	(None, 64, 64, 32)	2048
['block1a_se_excite[0][0]']		
2D)		
block1a_project_bn (BatchN	(None, 64, 64, 32)	128
['block1a_project_conv[0][0]']		
ormalization)		
block1b_dwconv (DepthwiseC	(None, 64, 64, 32)	288
['block1a_project_bn[0][0]']		
onv2D)		
block1b_bn (BatchNormaliza	(None, 64, 64, 32)	128
['block1b_dwconv[0][0]']		
tion)		
block1b_activation (Activa	(None, 64, 64, 32)	0
['block1b_bn[0][0]']		
tion)		
block1b_se_squeeze (Global	(None, 32)	0
['block1b_activation[0][0]']		
AveragePooling2D)		
block1b_se_reshape (Reshap	(None, 1, 1, 32)	0
['block1b_se_squeeze[0][0]']		
e)		
block1b_se_reduce (Conv2D)	(None, 1, 1, 8)	264
['block1b_se_reshape[0][0]']		
block1b_se_expand (Conv2D)	(None, 1, 1, 32)	288
['block1b_se_reduce[0][0]']		
block1b_se_excite (Multipl	(None, 64, 64, 32)	0
['block1b_activation[0][0]',		

```

y)
'block1b_se_expand[0][0]']

block1b_project_conv (Conv (None, 64, 64, 32) 1024
['block1b_se_excite[0][0]']
2D)

block1b_project_bn (BatchN (None, 64, 64, 32) 128
['block1b_project_conv[0][0]']
ormalization)

block1b_drop (Dropout) (None, 64, 64, 32) 0
['block1b_project_bn[0][0]']

block1b_add (Add) (None, 64, 64, 32) 0
['block1b_drop[0][0]'],
'block1a_project_bn[0][0]']

block1c_dwconv (DepthwiseC (None, 64, 64, 32) 288
['block1b_add[0][0]']
onv2D)

block1c_bn (BatchNormaliza (None, 64, 64, 32) 128
['block1c_dwconv[0][0]']
tion)

block1c_activation (Activa (None, 64, 64, 32) 0
['block1c_bn[0][0]']
tion)

block1c_se_squeeze (Global (None, 32) 0
['block1c_activation[0][0]']
AveragePooling2D)

block1c_se_reshape (Reshap (None, 1, 1, 32) 0
['block1c_se_squeeze[0][0]']
e)

block1c_se_reduce (Conv2D) (None, 1, 1, 8) 264
['block1c_se_reshape[0][0]']

block1c_se_expand (Conv2D) (None, 1, 1, 32) 288
['block1c_se_reduce[0][0]']

block1c_se_excite (Multipl (None, 64, 64, 32) 0
['block1c_activation[0][0]'],
y)
'block1c_se_expand[0][0]']

```

block1c_project_conv (Conv ['block1c_se_excite[0][0]'] 2D)	(None, 64, 64, 32)	1024
block1c_project_bn (BatchN ['block1c_project_conv[0][0]'] ormalization)	(None, 64, 64, 32)	128
block1c_drop (Dropout) ['block1c_project_bn[0][0]']	(None, 64, 64, 32)	0
block1c_add (Add) ['block1c_drop[0][0]', 'block1b_add[0][0]']	(None, 64, 64, 32)	0
block1d_dwconv (DepthwiseC ['block1c_add[0][0]'] onv2D)	(None, 64, 64, 32)	288
block1d_bn (BatchNormaliza ['block1d_dwconv[0][0]'] tion)	(None, 64, 64, 32)	128
block1d_activation (Activa ['block1d_bn[0][0]'] tion)	(None, 64, 64, 32)	0
block1d_se_squeeze (Global ['block1d_activation[0][0]'] AveragePooling2D)	(None, 32)	0
block1d_se_reshape (Reshap ['block1d_se_squeeze[0][0]'] e)	(None, 1, 1, 32)	0
block1d_se_reduce (Conv2D) ['block1d_se_reshape[0][0]']	(None, 1, 1, 8)	264
block1d_se_expand (Conv2D) ['block1d_se_reduce[0][0]']	(None, 1, 1, 32)	288
block1d_se_excite (Multipl ['block1d_activation[0][0]', y) 'block1d_se_expand[0][0]']	(None, 64, 64, 32)	0
block1d_project_conv (Conv	(None, 64, 64, 32)	1024

['block1d_se_excite[0][0]'] 2D)		
block1d_project_bn (BatchN (None, 64, 64, 32) ['block1d_project_conv[0][0]'] ormalization)	128	
block1d_drop (Dropout) (None, 64, 64, 32) ['block1d_project_bn[0][0]']	0	
block1d_add (Add) (None, 64, 64, 32) ['block1d_drop[0][0]', 'block1c_add[0][0]']	0	
block2a_expand_conv (Conv2 (None, 64, 64, 192) ['block1d_add[0][0]'] D)	6144	
block2a_expand_bn (BatchNo (None, 64, 64, 192) ['block2a_expand_conv[0][0]'] rmalization)	768	
block2a_expand_activation (None, 64, 64, 192) ['block2a_expand_bn[0][0]'] (Activation)	0	
block2a_dwconv_pad (ZeroPa (None, 65, 65, 192) ['block2a_expand_activation[0] dding2D)	0	[0]']
block2a_dwconv (DepthwiseC (None, 32, 32, 192) ['block2a_dwconv_pad[0][0]'] onv2D)	1728	
block2a_bn (BatchNormaliza (None, 32, 32, 192) ['block2a_dwconv[0][0]'] tion)	768	
block2a_activation (Activa (None, 32, 32, 192) ['block2a_bn[0][0]'] tion)	0	
block2a_se_squeeze (Global (None, 192) ['block2a_activation[0][0]'] AveragePooling2D)	0	
block2a_se_reshape (Reshap (None, 1, 1, 192) ['block2a_se_squeeze[0][0]']	0	

e)

block2a_se_reduce (Conv2D)	(None, 1, 1, 8)	1544
['block2a_se_reshape[0][0]']		
block2a_se_expand (Conv2D)	(None, 1, 1, 192)	1728
['block2a_se_reduce[0][0]']		
block2a_se_excite (Multipl	(None, 32, 32, 192)	0
['block2a_activation[0][0]',		
y)		
'block2a_se_expand[0][0]']		
block2a_project_conv (Conv	(None, 32, 32, 48)	9216
['block2a_se_excite[0][0]']		
2D)		
block2a_project_bn (BatchN	(None, 32, 32, 48)	192
['block2a_project_conv[0][0]']		
ormalization)		
block2b_expand_conv (Conv2	(None, 32, 32, 288)	13824
['block2a_project_bn[0][0]']		
D)		
block2b_expand_bn (BatchNo	(None, 32, 32, 288)	1152
['block2b_expand_conv[0][0]']		
rmalization)		
block2b_expand_activation	(None, 32, 32, 288)	0
['block2b_expand_bn[0][0]']		
(Activation)		
block2b_dwconv (DepthwiseC	(None, 32, 32, 288)	2592
['block2b_expand_activation[0]		
onv2D)		
		[0]']
block2b_bn (BatchNormaliza	(None, 32, 32, 288)	1152
['block2b_dwconv[0][0]']		
tion)		
block2b_activation (Activa	(None, 32, 32, 288)	0
['block2b_bn[0][0]']		
tion)		
block2b_se_squeeze (Global	(None, 288)	0
['block2b_activation[0][0]']		
AveragePooling2D)		

block2b_se_reshape (Reshape (None, 1, 1, 288) ['block2b_se_squeeze[0][0]'] e)	0	
block2b_se_reduce (Conv2D) (None, 1, 1, 12) ['block2b_se_reshape[0][0]']	3468	
block2b_se_expand (Conv2D) (None, 1, 1, 288) ['block2b_se_reduce[0][0]']	3744	
block2b_se_excite (Multipl (None, 32, 32, 288) ['block2b_activation[0][0]', y) 'block2b_se_expand[0][0]']	0	
block2b_project_conv (Conv (None, 32, 32, 48) ['block2b_se_excite[0][0]'] 2D)	13824	
block2b_project_bn (BatchN (None, 32, 32, 48) ['block2b_project_conv[0][0]'] ormalization)	192	
block2b_drop (Dropout) (None, 32, 32, 48) ['block2b_project_bn[0][0]']	0	
block2b_add (Add) (None, 32, 32, 48) ['block2b_drop[0][0]', 'block2a_project_bn[0][0]']	0	
block2c_expand_conv (Conv2 (None, 32, 32, 288) ['block2b_add[0][0]'] D)	13824	
block2c_expand_bn (BatchNo (None, 32, 32, 288) ['block2c_expand_conv[0][0]'] rmalization)	1152	
block2c_expand_activation (None, 32, 32, 288) ['block2c_expand_bn[0][0]'] (Activation)	0	
block2c_dwconv (DepthwiseC (None, 32, 32, 288) ['block2c_expand_activation[0] onv2D)	2592	[0]']
block2c_bn (BatchNormaliza (None, 32, 32, 288)	1152	

['block2c_dwconv[0][0]'] tion)		
block2c_activation (Activa ['block2c_bn[0][0]'] tion)	(None, 32, 32, 288)	0
block2c_se_squeeze (Global ['block2c_activation[0][0]'] AveragePooling2D)	(None, 288)	0
block2c_se_reshape (Reshap ['block2c_se_squeeze[0][0]'] e)	(None, 1, 1, 288)	0
block2c_se_reduce (Conv2D) ['block2c_se_reshape[0][0]']	(None, 1, 1, 12)	3468
block2c_se_expand (Conv2D) ['block2c_se_reduce[0][0]']	(None, 1, 1, 288)	3744
block2c_se_excite (Multipl ['block2c_activation[0][0]', y) 'block2c_se_expand[0][0]']	(None, 32, 32, 288)	0
block2c_project_conv (Conv ['block2c_se_excite[0][0]'] 2D)	(None, 32, 32, 48)	13824
block2c_project_bn (BatchN ['block2c_project_conv[0][0]'] ormalization)	(None, 32, 32, 48)	192
block2c_drop (Dropout) ['block2c_project_bn[0][0]']	(None, 32, 32, 48)	0
block2c_add (Add) ['block2c_drop[0][0]', 'block2b_add[0][0]']	(None, 32, 32, 48)	0
block2d_expand_conv (Conv2 ['block2c_add[0][0]'] D)	(None, 32, 32, 288)	13824
block2d_expand_bn (BatchNo ['block2d_expand_conv[0][0]'] rmalization)	(None, 32, 32, 288)	1152

block2d_expand_activation (None, 32, 32, 288) ['block2d_expand_bn[0][0]'] (Activation)	0	
block2d_dwconv (DepthwiseC (None, 32, 32, 288) ['block2d_expand_activation[0] onv2D)	2592	[0]']
block2d_bn (BatchNormaliza (None, 32, 32, 288) ['block2d_dwconv[0][0]'] tion)	1152	
block2d_activation (Activa (None, 32, 32, 288) ['block2d_bn[0][0]'] tion)	0	
block2d_se_squeeze (Global (None, 288) ['block2d_activation[0][0]'] AveragePooling2D)	0	
block2d_se_reshape (Reshap (None, 1, 1, 288) ['block2d_se_squeeze[0][0]'] e)	0	
block2d_se_reduce (Conv2D) (None, 1, 1, 12) ['block2d_se_reshape[0][0]']	3468	
block2d_se_expand (Conv2D) (None, 1, 1, 288) ['block2d_se_reduce[0][0]']	3744	
block2d_se_excite (Multipl (None, 32, 32, 288) ['block2d_activation[0][0]', y) 'block2d_se_expand[0][0]']	0	
block2d_project_conv (Conv (None, 32, 32, 48) ['block2d_se_excite[0][0]'] 2D)	13824	
block2d_project_bn (BatchN (None, 32, 32, 48) ['block2d_project_conv[0][0]'] ormalization)	192	
block2d_drop (Dropout) (None, 32, 32, 48) ['block2d_project_bn[0][0]']	0	
block2d_add (Add) (None, 32, 32, 48)	0	

```

['block2d_drop[0][0]',
'block2c_add[0][0]']

block2e_expand_conv (Conv2D (None, 32, 32, 288) 13824
['block2d_add[0][0]']
D)

block2e_expand_bn (BatchNormalization (None, 32, 32, 288) 1152
['block2e_expand_conv[0][0]']
rmalization)

block2e_expand_activation (None, 32, 32, 288) 0
['block2e_expand_bn[0][0]']
(Activation)

block2e_dwconv (DepthwiseConv2D (None, 32, 32, 288) 2592
['block2e_expand_activation[0]
onv2D) [0]']

block2e_bn (BatchNormalization (None, 32, 32, 288) 1152
['block2e_dwconv[0][0]']
tion)

block2e_activation (Activation (None, 32, 32, 288) 0
['block2e_bn[0][0]']
tion)

block2e_se_squeeze (GlobalAveragePooling2D (None, 288) 0
['block2e_activation[0][0]']
AveragePooling2D)

block2e_se_reshape (Reshape (None, 1, 1, 288) 0
['block2e_se_squeeze[0][0]']
e)

block2e_se_reduce (Conv2D (None, 1, 1, 12) 3468
['block2e_se_reshape[0][0]']

block2e_se_expand (Conv2D (None, 1, 1, 288) 3744
['block2e_se_reduce[0][0]']

block2e_se_excite (Multiply (None, 32, 32, 288) 0
['block2e_activation[0][0]',
y)
'block2e_se_expand[0][0]']

block2e_project_conv (Conv2D (None, 32, 32, 48) 13824
['block2e_se_excite[0][0]']

```

2D)		
block2e_project_bn (BatchNormaliza- tion) ['block2e_project_conv[0][0]']	(None, 32, 32, 48)	192
block2e_drop (Dropout) ['block2e_project_bn[0][0]']	(None, 32, 32, 48)	0
block2e_add (Add) ['block2e_drop[0][0]', 'block2d_add[0][0]']	(None, 32, 32, 48)	0
block2f_expand_conv (Conv2D) ['block2e_add[0][0]']	(None, 32, 32, 288)	13824
block2f_expand_bn (BatchNormaliza- tion) ['block2f_expand_conv[0][0]']	(None, 32, 32, 288)	1152
block2f_expand_activation (Activation) ['block2f_expand_bn[0][0]']	(None, 32, 32, 288)	0
block2f_dwconv (DepthwiseConv2D) ['block2f_expand_activation[0][0]']	(None, 32, 32, 288)	2592
block2f_bn (BatchNormalization) ['block2f_dwconv[0][0]']	(None, 32, 32, 288)	1152
block2f_activation (Activation) ['block2f_bn[0][0]']	(None, 32, 32, 288)	0
block2f_se_squeeze (GlobalAveragePooling2D) ['block2f_activation[0][0]']	(None, 288)	0
block2f_se_reshape (Reshape) ['block2f_se_squeeze[0][0]']	(None, 1, 1, 288)	0
block2f_se_reduce (Conv2D) ['block2f_se_reshape[0][0]']	(None, 1, 1, 12)	3468

block2f_se_expand (Conv2D) ['block2f_se_reduce[0][0]']	(None, 1, 1, 288)	3744	
block2f_se_excite (Multipl ['block2f_activation[0][0]', y) 'block2f_se_expand[0][0]']	(None, 32, 32, 288)	0	
block2f_project_conv (Conv ['block2f_se_excite[0][0]'] 2D)	(None, 32, 32, 48)	13824	
block2f_project_bn (BatchN ['block2f_project_conv[0][0]'] ormalization)	(None, 32, 32, 48)	192	
block2f_drop (Dropout) ['block2f_project_bn[0][0]']	(None, 32, 32, 48)	0	
block2f_add (Add) ['block2f_drop[0][0]', 'block2e_add[0][0]']	(None, 32, 32, 48)	0	
block2g_expand_conv (Conv2 ['block2f_add[0][0]'] D)	(None, 32, 32, 288)	13824	
block2g_expand_bn (BatchNo ['block2g_expand_conv[0][0]'] rmalization)	(None, 32, 32, 288)	1152	
block2g_expand_activation ['block2g_expand_bn[0][0]'] (Activation)	(None, 32, 32, 288)	0	
block2g_dwconv (DepthwiseC ['block2g_expand_activation[0] onv2D)	(None, 32, 32, 288)	2592	[0]']
block2g_bn (BatchNormaliza ['block2g_dwconv[0][0]'] tion)	(None, 32, 32, 288)	1152	
block2g_activation (Activa ['block2g_bn[0][0]'] tion)	(None, 32, 32, 288)	0	
block2g_se_squeeze (Global (None, 288)	(None, 288)	0	

['block2g_activation[0][0]'] AveragePooling2D)		
block2g_se_reshape (Reshape (None, 1, 1, 288) ['block2g_se_squeeze[0][0]'] e)		0
block2g_se_reduce (Conv2D) (None, 1, 1, 12) ['block2g_se_reshape[0][0]']		3468
block2g_se_expand (Conv2D) (None, 1, 1, 288) ['block2g_se_reduce[0][0]']		3744
block2g_se_excite (Multipl (None, 32, 32, 288) ['block2g_activation[0][0]', y) 'block2g_se_expand[0][0]']		0
block2g_project_conv (Conv (None, 32, 32, 48) ['block2g_se_excite[0][0]'] 2D)		13824
block2g_project_bn (BatchN (None, 32, 32, 48) ['block2g_project_conv[0][0]'] ormalization)		192
block2g_drop (Dropout) (None, 32, 32, 48) ['block2g_project_bn[0][0]']		0
block2g_add (Add) (None, 32, 32, 48) ['block2g_drop[0][0]', 'block2f_add[0][0]']		0
block3a_expand_conv (Conv2 (None, 32, 32, 288) ['block2g_add[0][0]'] D)		13824
block3a_expand_bn (BatchNo (None, 32, 32, 288) ['block3a_expand_conv[0][0]'] rmalization)		1152
block3a_expand_activation (None, 32, 32, 288) ['block3a_expand_bn[0][0]'] (Activation)		0
block3a_dwconv_pad (ZeroPa (None, 35, 35, 288) ['block3a_expand_activation[0] dding2D)		0

[0]']

block3a_dwconv (DepthwiseC ['block3a_dwconv_pad[0][0]' onv2D)	(None, 16, 16, 288)	7200
block3a_bn (BatchNormaliza ['block3a_dwconv[0][0]' tion)	(None, 16, 16, 288)	1152
block3a_activation (Activa ['block3a_bn[0][0]' tion)	(None, 16, 16, 288)	0
block3a_se_squeeze (Global ['block3a_activation[0][0]' AveragePooling2D)	(None, 288)	0
block3a_se_reshape (Reshap ['block3a_se_squeeze[0][0]' e)	(None, 1, 1, 288)	0
block3a_se_reduce (Conv2D) ['block3a_se_reshape[0][0]'	(None, 1, 1, 12)	3468
block3a_se_expand (Conv2D) ['block3a_se_reduce[0][0]'	(None, 1, 1, 288)	3744
block3a_se_excite (Multipl ['block3a_activation[0][0]' y) 'block3a_se_expand[0][0]'	(None, 16, 16, 288)	0
block3a_project_conv (Conv ['block3a_se_excite[0][0]' 2D)	(None, 16, 16, 80)	23040
block3a_project_bn (BatchN ['block3a_project_conv[0][0]' ormalization)	(None, 16, 16, 80)	320
block3b_expand_conv (Conv2 ['block3a_project_bn[0][0]' D)	(None, 16, 16, 480)	38400
block3b_expand_bn (BatchNo ['block3b_expand_conv[0][0]' rmalization)	(None, 16, 16, 480)	1920

block3b_expand_activation ['block3b_expand_bn[0][0]'] (Activation)	(None, 16, 16, 480)	0	
block3b_dwconv ['block3b_expand_activation[0] onv2D)	(DepthwiseC (None, 16, 16, 480)	12000	[0]']
block3b_bn ['block3b_dwconv[0][0]'] tion)	(BatchNormaliza (None, 16, 16, 480)	1920	
block3b_activation ['block3b_bn[0][0]'] tion)	(Activa (None, 16, 16, 480)	0	
block3b_se_squeeze ['block3b_activation[0][0]'] AveragePooling2D)	(Global (None, 480)	0	
block3b_se_reshape ['block3b_se_squeeze[0][0]'] e)	(Reshap (None, 1, 1, 480)	0	
block3b_se_reduce ['block3b_se_reshape[0][0]']	(Conv2D) (None, 1, 1, 20)	9620	
block3b_se_expand ['block3b_se_reduce[0][0]']	(Conv2D) (None, 1, 1, 480)	10080	
block3b_se_excite ['block3b_activation[0][0]', y) 'block3b_se_expand[0][0]']	(Multipl (None, 16, 16, 480)	0	
block3b_project_conv ['block3b_se_excite[0][0]'] 2D)	(Conv (None, 16, 16, 80)	38400	
block3b_project_bn ['block3b_project_conv[0][0]'] ormalization)	(BatchN (None, 16, 16, 80)	320	
block3b_drop ['block3b_project_bn[0][0]']	(Dropout) (None, 16, 16, 80)	0	
block3b_add ['block3b_drop[0][0]',	(Add) (None, 16, 16, 80)	0	

'block3a_project_bn[0][0]'		
block3c_expand_conv (Conv2 (None, 16, 16, 480)	38400	
['block3b_add[0][0]']		
D)		
block3c_expand_bn (BatchNo (None, 16, 16, 480)	1920	
['block3c_expand_conv[0][0]']		
rmalization)		
block3c_expand_activation (None, 16, 16, 480)	0	
['block3c_expand_bn[0][0]']		
(Activation)		
block3c_dwconv (DepthwiseC (None, 16, 16, 480)	12000	
['block3c_expand_activation[0]		
onv2D)		[0]']
block3c_bn (BatchNormaliza (None, 16, 16, 480)	1920	
['block3c_dwconv[0][0]']		
tion)		
block3c_activation (Activa (None, 16, 16, 480)	0	
['block3c_bn[0][0]']		
tion)		
block3c_se_squeeze (Global (None, 480)	0	
['block3c_activation[0][0]']		
AveragePooling2D)		
block3c_se_reshape (Reshap (None, 1, 1, 480)	0	
['block3c_se_squeeze[0][0]']		
e)		
block3c_se_reduce (Conv2D) (None, 1, 1, 20)	9620	
['block3c_se_reshape[0][0]']		
block3c_se_expand (Conv2D) (None, 1, 1, 480)	10080	
['block3c_se_reduce[0][0]']		
block3c_se_excite (Multipl (None, 16, 16, 480)	0	
['block3c_activation[0][0]']		
y)		
'block3c_se_expand[0][0]']		
block3c_project_conv (Conv (None, 16, 16, 80)	38400	
['block3c_se_excite[0][0]']		
2D)		

block3c_project_bn (BatchNormaliza ['block3c_project_conv[0][0]'] ormalization)	(None, 16, 16, 80)	320	
block3c_drop (Dropout) ['block3c_project_bn[0][0]']	(None, 16, 16, 80)	0	
block3c_add (Add) ['block3c_drop[0][0]', 'block3b_add[0][0]']	(None, 16, 16, 80)	0	
block3d_expand_conv (Conv2D) ['block3c_add[0][0]']	(None, 16, 16, 480)	38400	
block3d_expand_bn (BatchNormaliza ['block3d_expand_conv[0][0]'] rmalization)	(None, 16, 16, 480)	1920	
block3d_expand_activation ['block3d_expand_bn[0][0]'] (Activation)	(None, 16, 16, 480)	0	
block3d_dwconv (DepthwiseConv2D) ['block3d_expand_activation[0] onv2D)']	(None, 16, 16, 480)	12000	
block3d_bn (BatchNormalization) ['block3d_dwconv[0][0]']	(None, 16, 16, 480)	1920	
block3d_activation (Activation) ['block3d_bn[0][0]']	(None, 16, 16, 480)	0	
block3d_se_squeeze (GlobalAveragePooling2D) ['block3d_activation[0][0]']	(None, 480)	0	
block3d_se_reshape (Reshape) ['block3d_se_squeeze[0][0]'] e)	(None, 1, 1, 480)	0	
block3d_se_reduce (Conv2D) ['block3d_se_reshape[0][0]']	(None, 1, 1, 20)	9620	
block3d_se_expand (Conv2D)	(None, 1, 1, 480)	10080	

['block3d_se_reduce[0][0]']		
block3d_se_excite (Multipl (None, 16, 16, 480)	0	
['block3d_activation[0][0]', y) 'block3d_se_expand[0][0]']		
block3d_project_conv (Conv (None, 16, 16, 80)	38400	
['block3d_se_excite[0][0]'] 2D)		
block3d_project_bn (BatchN (None, 16, 16, 80)	320	
['block3d_project_conv[0][0]'] ormalization)		
block3d_drop (Dropout) (None, 16, 16, 80)	0	
['block3d_project_bn[0][0]']		
block3d_add (Add) (None, 16, 16, 80)	0	
['block3d_drop[0][0]', 'block3c_add[0][0]']		
block3e_expand_conv (Conv2 (None, 16, 16, 480)	38400	
['block3d_add[0][0]'] D)		
block3e_expand_bn (BatchNo (None, 16, 16, 480)	1920	
['block3e_expand_conv[0][0]'] rmalization)		
block3e_expand_activation (None, 16, 16, 480)	0	
['block3e_expand_bn[0][0]'] (Activation)		
block3e_dwconv (DepthwiseC (None, 16, 16, 480)	12000	
['block3e_expand_activation[0] onv2D)		[0]']
block3e_bn (BatchNormaliza (None, 16, 16, 480)	1920	
['block3e_dwconv[0][0]'] tion)		
block3e_activation (Activa (None, 16, 16, 480)	0	
['block3e_bn[0][0]'] tion)		
block3e_se_squeeze (Global (None, 480)	0	
['block3e_activation[0][0]']		

AveragePooling2D)		
block3e_se_reshape (Reshape ['block3e_se_squeeze[0][0]'] e)	(None, 1, 1, 480)	0
block3e_se_reduce (Conv2D ['block3e_se_reshape[0][0]']	(None, 1, 1, 20)	9620
block3e_se_expand (Conv2D ['block3e_se_reduce[0][0]']	(None, 1, 1, 480)	10080
block3e_se_excite (Multipl ['block3e_activation[0][0]', y) 'block3e_se_expand[0][0]']	(None, 16, 16, 480)	0
block3e_project_conv (Conv ['block3e_se_excite[0][0]'] 2D)	(None, 16, 16, 80)	38400
block3e_project_bn (BatchN ['block3e_project_conv[0][0]'] ormalization)	(None, 16, 16, 80)	320
block3e_drop (Dropout) ['block3e_project_bn[0][0]']	(None, 16, 16, 80)	0
block3e_add (Add) ['block3e_drop[0][0]', 'block3d_add[0][0]']	(None, 16, 16, 80)	0
block3f_expand_conv (Conv2 ['block3e_add[0][0]'] D)	(None, 16, 16, 480)	38400
block3f_expand_bn (BatchNo ['block3f_expand_conv[0][0]'] rmalization)	(None, 16, 16, 480)	1920
block3f_expand_activation ['block3f_expand_bn[0][0]'] (Activation)	(None, 16, 16, 480)	0
block3f_dwconv (DepthwiseC ['block3f_expand_activation[0] onv2D)	(None, 16, 16, 480)	12000

[0]']

block3f_bn (BatchNormaliza ['block3f_dwconv[0][0]' tion)	(None, 16, 16, 480)	1920
block3f_activation (Activa ['block3f_bn[0][0]' tion)	(None, 16, 16, 480)	0
block3f_se_squeeze (Global ['block3f_activation[0][0]' AveragePooling2D)	(None, 480)	0
block3f_se_reshape (Reshap ['block3f_se_squeeze[0][0]' e)	(None, 1, 1, 480)	0
block3f_se_reduce (Conv2D) ['block3f_se_reshape[0][0]']	(None, 1, 1, 20)	9620
block3f_se_expand (Conv2D) ['block3f_se_reduce[0][0]']	(None, 1, 1, 480)	10080
block3f_se_excite (Multipl ['block3f_activation[0][0]' y) 'block3f_se_expand[0][0]']	(None, 16, 16, 480)	0
block3f_project_conv (Conv ['block3f_se_excite[0][0]' 2D)	(None, 16, 16, 80)	38400
block3f_project_bn (BatchN ['block3f_project_conv[0][0]' ormalization)	(None, 16, 16, 80)	320
block3f_drop (Dropout) ['block3f_project_bn[0][0]']	(None, 16, 16, 80)	0
block3f_add (Add) ['block3f_drop[0][0]' 'block3e_add[0][0]']	(None, 16, 16, 80)	0
block3g_expand_conv (Conv2 ['block3f_add[0][0]'] D)	(None, 16, 16, 480)	38400
block3g_expand_bn (BatchNo ['block3g_expand_conv[0][0]']	(None, 16, 16, 480)	1920

rmalization)		
block3g_expand_activation ['block3g_expand_bn[0][0]'] (Activation)	(None, 16, 16, 480)	0
block3g_dwconv (DepthwiseC ['block3g_expand_activation[0] onv2D)	(None, 16, 16, 480)	12000 [0]']
block3g_bn (BatchNormaliza ['block3g_dwconv[0][0]'] tion)	(None, 16, 16, 480)	1920
block3g_activation (Activa ['block3g_bn[0][0]'] tion)	(None, 16, 16, 480)	0
block3g_se_squeeze (Global ['block3g_activation[0][0]'] AveragePooling2D)	(None, 480)	0
block3g_se_reshape (Reshap ['block3g_se_squeeze[0][0]'] e)	(None, 1, 1, 480)	0
block3g_se_reduce (Conv2D) ['block3g_se_reshape[0][0]']	(None, 1, 1, 20)	9620
block3g_se_expand (Conv2D) ['block3g_se_reduce[0][0]']	(None, 1, 1, 480)	10080
block3g_se_excite (Multipl ['block3g_activation[0][0]'], y) 'block3g_se_expand[0][0]']	(None, 16, 16, 480)	0
block3g_project_conv (Conv ['block3g_se_excite[0][0]'] 2D)	(None, 16, 16, 80)	38400
block3g_project_bn (BatchN ['block3g_project_conv[0][0]'] ormalization)	(None, 16, 16, 80)	320
block3g_drop (Dropout) ['block3g_project_bn[0][0]']	(None, 16, 16, 80)	0

block3g_add (Add)	(None, 16, 16, 80)	0
['block3g_drop[0][0]', 'block3f_add[0][0]']		
block4a_expand_conv (Conv2D)	(None, 16, 16, 480)	38400
['block3g_add[0][0]']		
block4a_expand_bn (BatchNormalization)	(None, 16, 16, 480)	1920
['block4a_expand_conv[0][0]']		
block4a_expand_activation (Activation)	(None, 16, 16, 480)	0
['block4a_expand_bn[0][0]']		
block4a_dwconv_pad (ZeroPadding2D)	(None, 17, 17, 480)	0
['block4a_expand_activation[0][0]']		
block4a_dwconv (DepthwiseConv2D)	(None, 8, 8, 480)	4320
['block4a_dwconv_pad[0][0]']		
block4a_bn (BatchNormalization)	(None, 8, 8, 480)	1920
['block4a_dwconv[0][0]']		
block4a_activation (Activation)	(None, 8, 8, 480)	0
['block4a_bn[0][0]']		
block4a_se_squeeze (GlobalAveragePooling2D)	(None, 480)	0
['block4a_activation[0][0]']		
block4a_se_reshape (Reshape)	(None, 1, 1, 480)	0
['block4a_se_squeeze[0][0]']		
block4a_se_reduce (Conv2D)	(None, 1, 1, 20)	9620
['block4a_se_reshape[0][0]']		
block4a_se_expand (Conv2D)	(None, 1, 1, 480)	10080
['block4a_se_reduce[0][0]']		
block4a_se_excite (Multiply)	(None, 8, 8, 480)	0
['block4a_activation[0][0]',		


```

y)
'block4a_se_expand[0][0]']

block4a_project_conv (Conv (None, 8, 8, 160) 76800
['block4a_se_excite[0][0]']
2D)

block4a_project_bn (BatchN (None, 8, 8, 160) 640
['block4a_project_conv[0][0]']
ormalization)

block4b_expand_conv (Conv2 (None, 8, 8, 960) 153600
['block4a_project_bn[0][0]']
D)

block4b_expand_bn (BatchNo (None, 8, 8, 960) 3840
['block4b_expand_conv[0][0]']
rmalization)

block4b_expand_activation (None, 8, 8, 960) 0
['block4b_expand_bn[0][0]']
(Activation)

block4b_dwconv (DepthwiseC (None, 8, 8, 960) 8640
['block4b_expand_activation[0]
onv2D) [0]']

block4b_bn (BatchNormaliza (None, 8, 8, 960) 3840
['block4b_dwconv[0][0]']
tion)

block4b_activation (Activa (None, 8, 8, 960) 0
['block4b_bn[0][0]']
tion)

block4b_se_squeeze (Global (None, 960) 0
['block4b_activation[0][0]']
AveragePooling2D)

block4b_se_reshape (Reshap (None, 1, 1, 960) 0
['block4b_se_squeeze[0][0]']
e)

block4b_se_reduce (Conv2D) (None, 1, 1, 40) 38440
['block4b_se_reshape[0][0]']

block4b_se_expand (Conv2D) (None, 1, 1, 960) 39360
['block4b_se_reduce[0][0]']

```

block4b_se_excite (Multipl ['block4b_activation[0][0]', y) 'block4b_se_expand[0][0]')	(None, 8, 8, 960)	0	
block4b_project_conv (Conv ['block4b_se_excite[0][0]'] 2D)	(None, 8, 8, 160)	153600	
block4b_project_bn (BatchN ['block4b_project_conv[0][0]'] ormalization)	(None, 8, 8, 160)	640	
block4b_drop (Dropout) ['block4b_project_bn[0][0]']	(None, 8, 8, 160)	0	
block4b_add (Add) ['block4b_drop[0][0]', 'block4a_project_bn[0][0]']	(None, 8, 8, 160)	0	
block4c_expand_conv (Conv2 ['block4b_add[0][0]'] D)	(None, 8, 8, 960)	153600	
block4c_expand_bn (BatchNo ['block4c_expand_conv[0][0]'] rmalization)	(None, 8, 8, 960)	3840	
block4c_expand_activation ['block4c_expand_bn[0][0]'] (Activation)	(None, 8, 8, 960)	0	
block4c_dwconv (DepthwiseC ['block4c_expand_activation[0] onv2D)	(None, 8, 8, 960)	8640	[0]'
block4c_bn (BatchNormaliza ['block4c_dwconv[0][0]'] tion)	(None, 8, 8, 960)	3840	
block4c_activation (Activa ['block4c_bn[0][0]'] tion)	(None, 8, 8, 960)	0	
block4c_se_squeeze (Global ['block4c_activation[0][0]'] AveragePooling2D)	(None, 960)	0	

block4c_se_reshape (Reshape ['block4c_se_squeeze[0][0]'] e)	(None, 1, 1, 960)	0
block4c_se_reduce (Conv2D ['block4c_se_reshape[0][0]']	(None, 1, 1, 40)	38440
block4c_se_expand (Conv2D ['block4c_se_reduce[0][0]']	(None, 1, 1, 960)	39360
block4c_se_excite (Multipl ['block4c_activation[0][0]', y] 'block4c_se_expand[0][0]']	(None, 8, 8, 960)	0
block4c_project_conv (Conv ['block4c_se_excite[0][0]'] 2D)	(None, 8, 8, 160)	153600
block4c_project_bn (BatchN ['block4c_project_conv[0][0]'] ormalization)	(None, 8, 8, 160)	640
block4c_drop (Dropout) ['block4c_project_bn[0][0]']	(None, 8, 8, 160)	0
block4c_add (Add) ['block4c_drop[0][0]', 'block4b_add[0][0]']	(None, 8, 8, 160)	0
block4d_expand_conv (Conv2 ['block4c_add[0][0]'] D)	(None, 8, 8, 960)	153600
block4d_expand_bn (BatchNo ['block4d_expand_conv[0][0]'] rmalization)	(None, 8, 8, 960)	3840
block4d_expand_activation ['block4d_expand_bn[0][0]'] (Activation)	(None, 8, 8, 960)	0
block4d_dwconv (DepthwiseC ['block4d_expand_activation[0] onv2D)	(None, 8, 8, 960)	8640
block4d_bn (BatchNormaliza	(None, 8, 8, 960)	3840

['block4d_dwconv[0][0]'] tion)		
block4d_activation (Activa (None, 8, 8, 960) ['block4d_bn[0][0]'] tion)		0
block4d_se_squeeze (Global (None, 960) ['block4d_activation[0][0]'] AveragePooling2D)		0
block4d_se_reshape (Reshap (None, 1, 1, 960) ['block4d_se_squeeze[0][0]'] e)		0
block4d_se_reduce (Conv2D) (None, 1, 1, 40) ['block4d_se_reshape[0][0]']		38440
block4d_se_expand (Conv2D) (None, 1, 1, 960) ['block4d_se_reduce[0][0]']		39360
block4d_se_excite (Multipl (None, 8, 8, 960) ['block4d_activation[0][0]', y) 'block4d_se_expand[0][0]']		0
block4d_project_conv (Conv (None, 8, 8, 160) ['block4d_se_excite[0][0]'] 2D)		153600
block4d_project_bn (BatchN (None, 8, 8, 160) ['block4d_project_conv[0][0]'] ormalization)		640
block4d_drop (Dropout) (None, 8, 8, 160) ['block4d_project_bn[0][0]']		0
block4d_add (Add) (None, 8, 8, 160) ['block4d_drop[0][0]', 'block4c_add[0][0]']		0
block4e_expand_conv (Conv2 (None, 8, 8, 960) ['block4d_add[0][0]'] D)		153600
block4e_expand_bn (BatchNo (None, 8, 8, 960) ['block4e_expand_conv[0][0]'] rmalization)		3840

block4e_expand_activation ['block4e_expand_bn[0][0]'] (Activation)	(None, 8, 8, 960)	0	
block4e_dwconv ['block4e_expand_activation[0] onv2D)	(DepthwiseC (None, 8, 8, 960)	8640	[0]']
block4e_bn ['block4e_dwconv[0][0]'] tion)	(BatchNormaliza (None, 8, 8, 960)	3840	
block4e_activation ['block4e_bn[0][0]'] tion)	(Activa (None, 8, 8, 960)	0	
block4e_se_squeeze ['block4e_activation[0][0]'] AveragePooling2D)	(Global (None, 960)	0	
block4e_se_reshape ['block4e_se_squeeze[0][0]'] e)	(Reshap (None, 1, 1, 960)	0	
block4e_se_reduce ['block4e_se_reshape[0][0]']	(Conv2D) (None, 1, 1, 40)	38440	
block4e_se_expand ['block4e_se_reduce[0][0]']	(Conv2D) (None, 1, 1, 960)	39360	
block4e_se_excite ['block4e_activation[0][0]'], y) 'block4e_se_expand[0][0]']	(Multipl (None, 8, 8, 960)	0	
block4e_project_conv ['block4e_se_excite[0][0]'] 2D)	(Conv (None, 8, 8, 160)	153600	
block4e_project_bn ['block4e_project_conv[0][0]'] ormalization)	(BatchN (None, 8, 8, 160)	640	
block4e_drop ['block4e_project_bn[0][0]']	(Dropout) (None, 8, 8, 160)	0	
block4e_add (Add)	(None, 8, 8, 160)	0	

```

['block4e_drop[0][0]',
'block4d_add[0][0]']

block4f_expand_conv (Conv2D (None, 8, 8, 960) 153600
['block4e_add[0][0]']
D)

block4f_expand_bn (BatchNormalization (None, 8, 8, 960) 3840
['block4f_expand_conv[0][0]']
rmalization)

block4f_expand_activation (None, 8, 8, 960) 0
['block4f_expand_bn[0][0]']
(Activation)

block4f_dwconv (DepthwiseConv2D (None, 8, 8, 960) 8640
['block4f_expand_activation[0]
onv2D) [0]']

block4f_bn (BatchNormalization (None, 8, 8, 960) 3840
['block4f_dwconv[0][0]']
tion)

block4f_activation (Activation (None, 8, 8, 960) 0
['block4f_bn[0][0]']
tion)

block4f_se_squeeze (GlobalAveragePooling2D (None, 960) 0
['block4f_activation[0][0]']
AveragePooling2D)

block4f_se_reshape (Reshape (None, 1, 1, 960) 0
['block4f_se_squeeze[0][0]']
e)

block4f_se_reduce (Conv2D (None, 1, 1, 40) 38440
['block4f_se_reshape[0][0]']

block4f_se_expand (Conv2D (None, 1, 1, 960) 39360
['block4f_se_reduce[0][0]']

block4f_se_excite (Multiply (None, 8, 8, 960) 0
['block4f_activation[0][0]',
y)
'block4f_se_expand[0][0]']

block4f_project_conv (Conv2D (None, 8, 8, 160) 153600
['block4f_se_excite[0][0]']

```

2D)		
block4f_project_bn (BatchNormaliza- tion) ['block4f_project_conv[0][0]']	(None, 8, 8, 160)	640
block4f_drop (Dropout) ['block4f_project_bn[0][0]']	(None, 8, 8, 160)	0
block4f_add (Add) ['block4f_drop[0][0]', 'block4e_add[0][0]']	(None, 8, 8, 160)	0
block4g_expand_conv (Conv2D) ['block4f_add[0][0]']	(None, 8, 8, 960)	153600
block4g_expand_bn (BatchNormaliza- tion) ['block4g_expand_conv[0][0]']	(None, 8, 8, 960)	3840
block4g_expand_activation (Activation) ['block4g_expand_bn[0][0]']	(None, 8, 8, 960)	0
block4g_dwconv (DepthwiseConv2D) ['block4g_expand_activation[0][0]']	(None, 8, 8, 960)	8640
block4g_bn (BatchNormalization) ['block4g_dwconv[0][0]']	(None, 8, 8, 960)	3840
block4g_activation (Activation) ['block4g_bn[0][0]']	(None, 8, 8, 960)	0
block4g_se_squeeze (GlobalAveragePooling2D) ['block4g_activation[0][0]']	(None, 960)	0
block4g_se_reshape (Reshape) ['block4g_se_squeeze[0][0]']	(None, 1, 1, 960)	0
block4g_se_reduce (Conv2D) ['block4g_se_reshape[0][0]']	(None, 1, 1, 40)	38440

block4g_se_expand (Conv2D) ['block4g_se_reduce[0][0]']	(None, 1, 1, 960)	39360
block4g_se_excite (Multipl ['block4g_activation[0][0]', y) 'block4g_se_expand[0][0]']	(None, 8, 8, 960)	0
block4g_project_conv (Conv ['block4g_se_excite[0][0]'] 2D)	(None, 8, 8, 160)	153600
block4g_project_bn (BatchN ['block4g_project_conv[0][0]'] ormalization)	(None, 8, 8, 160)	640
block4g_drop (Dropout) ['block4g_project_bn[0][0]']	(None, 8, 8, 160)	0
block4g_add (Add) ['block4g_drop[0][0]', 'block4f_add[0][0]']	(None, 8, 8, 160)	0
block4h_expand_conv (Conv2 ['block4g_add[0][0]'] D)	(None, 8, 8, 960)	153600
block4h_expand_bn (BatchNo ['block4h_expand_conv[0][0]'] rmalization)	(None, 8, 8, 960)	3840
block4h_expand_activation ['block4h_expand_bn[0][0]'] (Activation)	(None, 8, 8, 960)	0
block4h_dwconv (DepthwiseC ['block4h_expand_activation[0] onv2D)	(None, 8, 8, 960)	8640
block4h_bn (BatchNormaliza ['block4h_dwconv[0][0]'] tion)	(None, 8, 8, 960)	3840
block4h_activation (Activa ['block4h_bn[0][0]'] tion)	(None, 8, 8, 960)	0
block4h_se_squeeze (Global (None, 960)		0

['block4h_activation[0][0]'] AveragePooling2D)		
block4h_se_reshape (Reshape (None, 1, 1, 960) ['block4h_se_squeeze[0][0]'] e)		0
block4h_se_reduce (Conv2D) (None, 1, 1, 40) ['block4h_se_reshape[0][0]']		38440
block4h_se_expand (Conv2D) (None, 1, 1, 960) ['block4h_se_reduce[0][0]']		39360
block4h_se_excite (Multipl (None, 8, 8, 960) ['block4h_activation[0][0]', y) 'block4h_se_expand[0][0]']		0
block4h_project_conv (Conv (None, 8, 8, 160) ['block4h_se_excite[0][0]'] 2D)		153600
block4h_project_bn (BatchN (None, 8, 8, 160) ['block4h_project_conv[0][0]'] ormalization)		640
block4h_drop (Dropout) (None, 8, 8, 160) ['block4h_project_bn[0][0]']		0
block4h_add (Add) (None, 8, 8, 160) ['block4h_drop[0][0]', 'block4g_add[0][0]']		0
block4i_expand_conv (Conv2 (None, 8, 8, 960) ['block4h_add[0][0]'] D)		153600
block4i_expand_bn (BatchNo (None, 8, 8, 960) ['block4i_expand_conv[0][0]'] rmalization)		3840
block4i_expand_activation (None, 8, 8, 960) ['block4i_expand_bn[0][0]'] (Activation)		0
block4i_dwconv (DepthwiseC (None, 8, 8, 960) ['block4i_expand_activation[0] onv2D)		8640

[0]']

block4i_bn (BatchNormaliza ['block4i_dwconv[0][0]' tion)	(None, 8, 8, 960)	3840
block4i_activation (Activa ['block4i_bn[0][0]' tion)	(None, 8, 8, 960)	0
block4i_se_squeeze (Global ['block4i_activation[0][0]' AveragePooling2D)	(None, 960)	0
block4i_se_reshape (Reshap ['block4i_se_squeeze[0][0]' e)	(None, 1, 1, 960)	0
block4i_se_reduce (Conv2D) ['block4i_se_reshape[0][0]']	(None, 1, 1, 40)	38440
block4i_se_expand (Conv2D) ['block4i_se_reduce[0][0]']	(None, 1, 1, 960)	39360
block4i_se_excite (Multipl ['block4i_activation[0][0]' y) 'block4i_se_expand[0][0]']	(None, 8, 8, 960)	0
block4i_project_conv (Conv ['block4i_se_excite[0][0]' 2D)	(None, 8, 8, 160)	153600
block4i_project_bn (BatchN ['block4i_project_conv[0][0]' ormalization)	(None, 8, 8, 160)	640
block4i_drop (Dropout) ['block4i_project_bn[0][0]']	(None, 8, 8, 160)	0
block4i_add (Add) ['block4i_drop[0][0]' 'block4h_add[0][0]']	(None, 8, 8, 160)	0
block4j_expand_conv (Conv2 ['block4i_add[0][0]'] D)	(None, 8, 8, 960)	153600
block4j_expand_bn (BatchNo	(None, 8, 8, 960)	3840

['block4j_expand_conv[0][0]'] rmalization)		
block4j_expand_activation (None, 8, 8, 960) ['block4j_expand_bn[0][0]'] (Activation)	0	
block4j_dwconv (DepthwiseC (None, 8, 8, 960) ['block4j_expand_activation[0] onv2D)	8640	[0]']
block4j_bn (BatchNormaliza (None, 8, 8, 960) ['block4j_dwconv[0][0]'] tion)	3840	
block4j_activation (Activa (None, 8, 8, 960) ['block4j_bn[0][0]'] tion)	0	
block4j_se_squeeze (Global (None, 960) ['block4j_activation[0][0]'] AveragePooling2D)	0	
block4j_se_reshape (Reshap (None, 1, 1, 960) ['block4j_se_squeeze[0][0]'] e)	0	
block4j_se_reduce (Conv2D) (None, 1, 1, 40) ['block4j_se_reshape[0][0]']	38440	
block4j_se_expand (Conv2D) (None, 1, 1, 960) ['block4j_se_reduce[0][0]']	39360	
block4j_se_excite (Multipl (None, 8, 8, 960) ['block4j_activation[0][0]', y) 'block4j_se_expand[0][0]']	0	
block4j_project_conv (Conv (None, 8, 8, 160) ['block4j_se_excite[0][0]'] 2D)	153600	
block4j_project_bn (BatchN (None, 8, 8, 160) ['block4j_project_conv[0][0]'] ormalization)	640	
block4j_drop (Dropout) (None, 8, 8, 160) ['block4j_project_bn[0][0]']	0	

block4j_add (Add)	(None, 8, 8, 160)	0	
['block4j_drop[0][0]', 'block4i_add[0][0]']			
block5a_expand_conv (Conv2D)	(None, 8, 8, 960)	153600	
['block4j_add[0][0]']			
block5a_expand_bn (BatchNormalization)	(None, 8, 8, 960)	3840	
['block5a_expand_conv[0][0]']			
block5a_expand_activation (Activation)	(None, 8, 8, 960)	0	
['block5a_expand_bn[0][0]']			
block5a_dwconv (DepthwiseConv2D)	(None, 8, 8, 960)	24000	
['block5a_expand_activation[0][0]']			
block5a_bn (BatchNormalization)	(None, 8, 8, 960)	3840	
['block5a_dwconv[0][0]']			
block5a_activation (Activation)	(None, 8, 8, 960)	0	
['block5a_bn[0][0]']			
block5a_se_squeeze (GlobalAveragePooling2D)	(None, 960)	0	
['block5a_activation[0][0]']			
block5a_se_reshape (Reshape)	(None, 1, 1, 960)	0	
['block5a_se_squeeze[0][0]']			
block5a_se_reduce (Conv2D)	(None, 1, 1, 40)	38440	
['block5a_se_reshape[0][0]']			
block5a_se_expand (Conv2D)	(None, 1, 1, 960)	39360	
['block5a_se_reduce[0][0]']			
block5a_se_excite (Multiply)	(None, 8, 8, 960)	0	
['block5a_activation[0][0]', 'block5a_se_expand[0][0]']			

block5a_project_conv (Conv2D) ['block5a_se_excite[0][0]']	(None, 8, 8, 224)	215040	
block5a_project_bn (BatchNormalization) ['block5a_project_conv[0][0]']	(None, 8, 8, 224)	896	
block5b_expand_conv (Conv2D) ['block5a_project_bn[0][0]']	(None, 8, 8, 1344)	301056	
block5b_expand_bn (BatchNormalization) ['block5b_expand_conv[0][0]']	(None, 8, 8, 1344)	5376	
block5b_expand_activation (Activation) ['block5b_expand_bn[0][0]']	(None, 8, 8, 1344)	0	
block5b_dwconv (DepthwiseConv2D) ['block5b_expand_activation[0][0]']	(None, 8, 8, 1344)	33600	[0]'
block5b_bn (BatchNormalization) ['block5b_dwconv[0][0]']	(None, 8, 8, 1344)	5376	
block5b_activation (Activation) ['block5b_bn[0][0]']	(None, 8, 8, 1344)	0	
block5b_se_squeeze (GlobalAveragePooling2D) ['block5b_activation[0][0]']	(None, 1344)	0	
block5b_se_reshape (Reshape) ['block5b_se_squeeze[0][0]']	(None, 1, 1, 1344)	0	
block5b_se_reduce (Conv2D) ['block5b_se_reshape[0][0]']	(None, 1, 1, 56)	75320	
block5b_se_expand (Conv2D) ['block5b_se_reduce[0][0]']	(None, 1, 1, 1344)	76608	
block5b_se_excite (Multiply) ['block5b_activation[0][0]',	(None, 8, 8, 1344)	0	

```

y)
'block5b_se_expand[0][0]']

block5b_project_conv (Conv (None, 8, 8, 224) 301056
['block5b_se_excite[0][0]']
2D)

block5b_project_bn (BatchN (None, 8, 8, 224) 896
['block5b_project_conv[0][0]']
ormalization)

block5b_drop (Dropout) (None, 8, 8, 224) 0
['block5b_project_bn[0][0]']

block5b_add (Add) (None, 8, 8, 224) 0
['block5b_drop[0][0] ',
'block5a_project_bn[0][0]']

block5c_expand_conv (Conv2 (None, 8, 8, 1344) 301056
['block5b_add[0][0]']
D)

block5c_expand_bn (BatchNo (None, 8, 8, 1344) 5376
['block5c_expand_conv[0][0]']
rmalization)

block5c_expand_activation (None, 8, 8, 1344) 0
['block5c_expand_bn[0][0]']
(Activation)

block5c_dwconv (DepthwiseC (None, 8, 8, 1344) 33600
['block5c_expand_activation[0]
onv2D) [0]']

block5c_bn (BatchNormaliza (None, 8, 8, 1344) 5376
['block5c_dwconv[0][0]']
tion)

block5c_activation (Activa (None, 8, 8, 1344) 0
['block5c_bn[0][0]']
tion)

block5c_se_squeeze (Global (None, 1344) 0
['block5c_activation[0][0]']
AveragePooling2D)

block5c_se_reshape (Reshap (None, 1, 1, 1344) 0
['block5c_se_squeeze[0][0]']

```

e)

block5c_se_reduce (Conv2D)	(None, 1, 1, 56)	75320	
['block5c_se_reshape[0][0]']			
block5c_se_expand (Conv2D)	(None, 1, 1, 1344)	76608	
['block5c_se_reduce[0][0]']			
block5c_se_excite (Multipl	(None, 8, 8, 1344)	0	
['block5c_activation[0][0]',			
y)			
'block5c_se_expand[0][0]']			
block5c_project_conv (Conv	(None, 8, 8, 224)	301056	
['block5c_se_excite[0][0]']			
2D)			
block5c_project_bn (BatchN	(None, 8, 8, 224)	896	
['block5c_project_conv[0][0]']			
ormalization)			
block5c_drop (Dropout)	(None, 8, 8, 224)	0	
['block5c_project_bn[0][0]']			
block5c_add (Add)	(None, 8, 8, 224)	0	
['block5c_drop[0][0]',			
'block5b_add[0][0]']			
block5d_expand_conv (Conv2	(None, 8, 8, 1344)	301056	
['block5c_add[0][0]']			
D)			
block5d_expand_bn (BatchNo	(None, 8, 8, 1344)	5376	
['block5d_expand_conv[0][0]']			
rmalization)			
block5d_expand_activation	(None, 8, 8, 1344)	0	
['block5d_expand_bn[0][0]']			
(Activation)			
block5d_dwconv (DepthwiseC	(None, 8, 8, 1344)	33600	
['block5d_expand_activation[0]			
onv2D)			
			[0]']
block5d_bn (BatchNormaliza	(None, 8, 8, 1344)	5376	
['block5d_dwconv[0][0]']			
tion)			

block5d_activation (Activation) ['block5d_bn[0][0]']	(None, 8, 8, 1344)	0
block5d_se_squeeze (Global AveragePooling2D) ['block5d_activation[0][0]']	(None, 1344)	0
block5d_se_reshape (Reshape) ['block5d_se_squeeze[0][0]']	(None, 1, 1, 1344)	0
block5d_se_reduce (Conv2D) ['block5d_se_reshape[0][0]']	(None, 1, 1, 56)	75320
block5d_se_expand (Conv2D) ['block5d_se_reduce[0][0]']	(None, 1, 1, 1344)	76608
block5d_se_excite (Multiply) ['block5d_activation[0][0]', y] ['block5d_se_expand[0][0]']	(None, 8, 8, 1344)	0
block5d_project_conv (Conv2D) ['block5d_se_excite[0][0]']	(None, 8, 8, 224)	301056
block5d_project_bn (Batch Normalization) ['block5d_project_conv[0][0]']	(None, 8, 8, 224)	896
block5d_drop (Dropout) ['block5d_project_bn[0][0]']	(None, 8, 8, 224)	0
block5d_add (Add) ['block5d_drop[0][0]', ['block5c_add[0][0]']	(None, 8, 8, 224)	0
block5e_expand_conv (Conv2D) ['block5d_add[0][0]']	(None, 8, 8, 1344)	301056
block5e_expand_bn (Batch Normalization) ['block5e_expand_conv[0][0]']	(None, 8, 8, 1344)	5376
block5e_expand_activation (Activation) ['block5e_expand_bn[0][0]']	(None, 8, 8, 1344)	0

(Activation)		
block5e_dwconv (DepthwiseC (None, 8, 8, 1344)	33600	
['block5e_expand_activation[0]		[0]']
onv2D)		
block5e_bn (BatchNormaliza (None, 8, 8, 1344)	5376	
['block5e_dwconv[0] [0]']		
tion)		
block5e_activation (Activa (None, 8, 8, 1344)	0	
['block5e_bn[0] [0]']		
tion)		
block5e_se_squeeze (Global (None, 1344)	0	
['block5e_activation[0] [0]']		
AveragePooling2D)		
block5e_se_reshape (Reshap (None, 1, 1, 1344)	0	
['block5e_se_squeeze[0] [0]']		
e)		
block5e_se_reduce (Conv2D) (None, 1, 1, 56)	75320	
['block5e_se_reshape[0] [0]']		
block5e_se_expand (Conv2D) (None, 1, 1, 1344)	76608	
['block5e_se_reduce[0] [0]']		
block5e_se_excite (Multipl (None, 8, 8, 1344)	0	
['block5e_activation[0] [0]']		
y)		
'block5e_se_expand[0] [0]']		
block5e_project_conv (Conv (None, 8, 8, 224)	301056	
['block5e_se_excite[0] [0]']		
2D)		
block5e_project_bn (BatchN (None, 8, 8, 224)	896	
['block5e_project_conv[0] [0]']		
ormalization)		
block5e_drop (Dropout) (None, 8, 8, 224)	0	
['block5e_project_bn[0] [0]']		
block5e_add (Add) (None, 8, 8, 224)	0	
['block5e_drop[0] [0]']		
'block5d_add[0] [0]']		

block5f_expand_conv (Conv2 (None, 8, 8, 1344) ['block5e_add[0][0]'] D)	301056	
block5f_expand_bn (BatchNo (None, 8, 8, 1344) ['block5f_expand_conv[0][0]'] rmalization)	5376	
block5f_expand_activation (None, 8, 8, 1344) ['block5f_expand_bn[0][0]'] (Activation)	0	
block5f_dwconv (DepthwiseC (None, 8, 8, 1344) ['block5f_expand_activation[0] onv2D)	33600	[0]']
block5f_bn (BatchNormaliza (None, 8, 8, 1344) ['block5f_dwconv[0][0]'] tion)	5376	
block5f_activation (Activa (None, 8, 8, 1344) ['block5f_bn[0][0]'] tion)	0	
block5f_se_squeeze (Global (None, 1344) ['block5f_activation[0][0]'] AveragePooling2D)	0	
block5f_se_reshape (Reshap (None, 1, 1, 1344) ['block5f_se_squeeze[0][0]'] e)	0	
block5f_se_reduce (Conv2D) (None, 1, 1, 56) ['block5f_se_reshape[0][0]']	75320	
block5f_se_expand (Conv2D) (None, 1, 1, 1344) ['block5f_se_reduce[0][0]']	76608	
block5f_se_excite (Multipl (None, 8, 8, 1344) ['block5f_activation[0][0]', y) 'block5f_se_expand[0][0]']	0	
block5f_project_conv (Conv (None, 8, 8, 224) ['block5f_se_excite[0][0]'] 2D)	301056	
block5f_project_bn (BatchN (None, 8, 8, 224)	896	

['block5f_project_conv[0][0]'] ormalization)		
block5f_drop (Dropout) ['block5f_project_bn[0][0]']	(None, 8, 8, 224)	0
block5f_add (Add) ['block5f_drop[0][0]', 'block5e_add[0][0]']	(None, 8, 8, 224)	0
block5g_expand_conv (Conv2 ['block5f_add[0][0]'] D)	(None, 8, 8, 1344)	301056
block5g_expand_bn (BatchNo ['block5g_expand_conv[0][0]'] rmalization)	(None, 8, 8, 1344)	5376
block5g_expand_activation ['block5g_expand_bn[0][0]'] (Activation)	(None, 8, 8, 1344)	0
block5g_dwconv (DepthwiseC ['block5g_expand_activation[0] onv2D)	(None, 8, 8, 1344)	33600 [0]']
block5g_bn (BatchNormaliza ['block5g_dwconv[0][0]'] tion)	(None, 8, 8, 1344)	5376
block5g_activation (Activa ['block5g_bn[0][0]'] tion)	(None, 8, 8, 1344)	0
block5g_se_squeeze (Global ['block5g_activation[0][0]'] AveragePooling2D)	(None, 1344)	0
block5g_se_reshape (Reshap ['block5g_se_squeeze[0][0]'] e)	(None, 1, 1, 1344)	0
block5g_se_reduce (Conv2D) ['block5g_se_reshape[0][0]']	(None, 1, 1, 56)	75320
block5g_se_expand (Conv2D) ['block5g_se_reduce[0][0]']	(None, 1, 1, 1344)	76608

block5g_se_excite (Multipl ['block5g_activation[0][0]', y) 'block5g_se_expand[0][0]')	(None, 8, 8, 1344)	0	
block5g_project_conv (Conv ['block5g_se_excite[0][0]'] 2D)	(None, 8, 8, 224)	301056	
block5g_project_bn (BatchN ['block5g_project_conv[0][0]'] ormalization)	(None, 8, 8, 224)	896	
block5g_drop (Dropout) ['block5g_project_bn[0][0]']	(None, 8, 8, 224)	0	
block5g_add (Add) ['block5g_drop[0][0]', 'block5f_add[0][0]']	(None, 8, 8, 224)	0	
block5h_expand_conv (Conv2 ['block5g_add[0][0]'] D)	(None, 8, 8, 1344)	301056	
block5h_expand_bn (BatchNo ['block5h_expand_conv[0][0]'] rmalization)	(None, 8, 8, 1344)	5376	
block5h_expand_activation ['block5h_expand_bn[0][0]'] (Activation)	(None, 8, 8, 1344)	0	
block5h_dwconv (DepthwiseC ['block5h_expand_activation[0] onv2D)	(None, 8, 8, 1344)	33600	[0]']
block5h_bn (BatchNormaliza ['block5h_dwconv[0][0]'] tion)	(None, 8, 8, 1344)	5376	
block5h_activation (Activa ['block5h_bn[0][0]'] tion)	(None, 8, 8, 1344)	0	
block5h_se_squeeze (Global ['block5h_activation[0][0]'] AveragePooling2D)	(None, 1344)	0	

block5h_se_reshape (Reshape (None, 1, 1, 1344) ['block5h_se_squeeze[0][0]'] e)	0	
block5h_se_reduce (Conv2D (None, 1, 1, 56) ['block5h_se_reshape[0][0]']	75320	
block5h_se_expand (Conv2D (None, 1, 1, 1344) ['block5h_se_reduce[0][0]']	76608	
block5h_se_excite (Multiply (None, 8, 8, 1344) ['block5h_activation[0][0]', y) 'block5h_se_expand[0][0]']	0	
block5h_project_conv (Conv2D (None, 8, 8, 224) ['block5h_se_excite[0][0]'] 2D)	301056	
block5h_project_bn (BatchNormalization (None, 8, 8, 224) ['block5h_project_conv[0][0]'] ormalization)	896	
block5h_drop (Dropout (None, 8, 8, 224) ['block5h_project_bn[0][0]']	0	
block5h_add (Add (None, 8, 8, 224) ['block5h_drop[0][0]', 'block5g_add[0][0]']	0	
block5i_expand_conv (Conv2D (None, 8, 8, 1344) ['block5h_add[0][0]'] D)	301056	
block5i_expand_bn (BatchNormalization (None, 8, 8, 1344) ['block5i_expand_conv[0][0]'] ormalization)	5376	
block5i_expand_activation (None, 8, 8, 1344) ['block5i_expand_bn[0][0]'] (Activation)	0	
block5i_dwconv (DepthwiseConv2D (None, 8, 8, 1344) ['block5i_expand_activation[0][0]'] onv2D)	33600	[0]']
block5i_bn (BatchNormalization (None, 8, 8, 1344) ['block5i_dwconv[0][0]']	5376	

tion)		
block5i_activation (Activation) (None, 8, 8, 1344)	0	
['block5i_bn[0][0]']		
tion)		
block5i_se_squeeze (GlobalAveragePooling2D) (None, 1344)	0	
['block5i_activation[0][0]']		
AveragePooling2D)		
block5i_se_reshape (Reshape) (None, 1, 1, 1344)	0	
['block5i_se_squeeze[0][0]']		
e)		
block5i_se_reduce (Conv2D) (None, 1, 1, 56)	75320	
['block5i_se_reshape[0][0]']		
block5i_se_expand (Conv2D) (None, 1, 1, 1344)	76608	
['block5i_se_reduce[0][0]']		
block5i_se_excite (Multiply) (None, 8, 8, 1344)	0	
['block5i_activation[0][0]',		
y)		
'block5i_se_expand[0][0]']		
block5i_project_conv (Conv2D) (None, 8, 8, 224)	301056	
['block5i_se_excite[0][0]']		
2D)		
block5i_project_bn (BatchNormalization) (None, 8, 8, 224)	896	
['block5i_project_conv[0][0]']		
ormalization)		
block5i_drop (Dropout) (None, 8, 8, 224)	0	
['block5i_project_bn[0][0]']		
block5i_add (Add) (None, 8, 8, 224)	0	
['block5i_drop[0][0]',		
'block5h_add[0][0]']		
block5j_expand_conv (Conv2D) (None, 8, 8, 1344)	301056	
['block5i_add[0][0]']		
D)		
block5j_expand_bn (BatchNormalization) (None, 8, 8, 1344)	5376	
['block5j_expand_conv[0][0]']		
rmalization)		

block5j_expand_activation (None, 8, 8, 1344) ['block5j_expand_bn[0][0]'] (Activation)	0	
block5j_dwconv (DepthwiseC (None, 8, 8, 1344) ['block5j_expand_activation[0] onv2D)	33600	[0]']
block5j_bn (BatchNormaliza (None, 8, 8, 1344) ['block5j_dwconv[0][0]'] tion)	5376	
block5j_activation (Activa (None, 8, 8, 1344) ['block5j_bn[0][0]'] tion)	0	
block5j_se_squeeze (Global (None, 1344) ['block5j_activation[0][0]'] AveragePooling2D)	0	
block5j_se_reshape (Reshap (None, 1, 1, 1344) ['block5j_se_squeeze[0][0]'] e)	0	
block5j_se_reduce (Conv2D) (None, 1, 1, 56) ['block5j_se_reshape[0][0]']	75320	
block5j_se_expand (Conv2D) (None, 1, 1, 1344) ['block5j_se_reduce[0][0]']	76608	
block5j_se_excite (Multipl (None, 8, 8, 1344) ['block5j_activation[0][0]', y) 'block5j_se_expand[0][0]']	0	
block5j_project_conv (Conv (None, 8, 8, 224) ['block5j_se_excite[0][0]'] 2D)	301056	
block5j_project_bn (BatchN (None, 8, 8, 224) ['block5j_project_conv[0][0]'] ormalization)	896	
block5j_drop (Dropout) (None, 8, 8, 224) ['block5j_project_bn[0][0]']	0	
block5j_add (Add) (None, 8, 8, 224) ['block5j_drop[0][0]',	0	

'block5i_add[0][0]'		
block6a_expand_conv (Conv2 (None, 8, 8, 1344)	301056	
['block5j_add[0][0]'		
D)		
block6a_expand_bn (BatchNo (None, 8, 8, 1344)	5376	
['block6a_expand_conv[0][0]'		
rmalization)		
block6a_expand_activation (None, 8, 8, 1344)	0	
['block6a_expand_bn[0][0]'		
(Activation)		
block6a_dwconv_pad (ZeroPa (None, 11, 11, 1344)	0	
['block6a_expand_activation[0]		
dding2D)		[0]'
block6a_dwconv (DepthwiseC (None, 4, 4, 1344)	33600	
['block6a_dwconv_pad[0][0]'		
onv2D)		
block6a_bn (BatchNormaliza (None, 4, 4, 1344)	5376	
['block6a_dwconv[0][0]'		
tion)		
block6a_activation (Activa (None, 4, 4, 1344)	0	
['block6a_bn[0][0]'		
tion)		
block6a_se_squeeze (Global (None, 1344)	0	
['block6a_activation[0][0]'		
AveragePooling2D)		
block6a_se_reshape (Reshap (None, 1, 1, 1344)	0	
['block6a_se_squeeze[0][0]'		
e)		
block6a_se_reduce (Conv2D) (None, 1, 1, 56)	75320	
['block6a_se_reshape[0][0]'		
block6a_se_expand (Conv2D) (None, 1, 1, 1344)	76608	
['block6a_se_reduce[0][0]'		
block6a_se_excite (Multipl (None, 4, 4, 1344)	0	
['block6a_activation[0][0]'		
y)		
'block6a_se_expand[0][0]'		

block6a_project_conv (Conv (None, 4, 4, 384) ['block6a_se_excite[0][0]'] 2D)	516096	
block6a_project_bn (BatchN (None, 4, 4, 384) ['block6a_project_conv[0][0]'] ormalization)	1536	
block6b_expand_conv (Conv2 (None, 4, 4, 2304) ['block6a_project_bn[0][0]'] D)	884736	
block6b_expand_bn (BatchNo (None, 4, 4, 2304) ['block6b_expand_conv[0][0]'] rmalization)	9216	
block6b_expand_activation (None, 4, 4, 2304) ['block6b_expand_bn[0][0]'] (Activation)	0	
block6b_dwconv (DepthwiseC (None, 4, 4, 2304) ['block6b_expand_activation[0] onv2D)	57600	[0]']
block6b_bn (BatchNormaliza (None, 4, 4, 2304) ['block6b_dwconv[0][0]'] tion)	9216	
block6b_activation (Activa (None, 4, 4, 2304) ['block6b_bn[0][0]'] tion)	0	
block6b_se_squeeze (Global (None, 2304) ['block6b_activation[0][0]'] AveragePooling2D)	0	
block6b_se_reshape (Reshap (None, 1, 1, 2304) ['block6b_se_squeeze[0][0]'] e)	0	
block6b_se_reduce (Conv2D) (None, 1, 1, 96) ['block6b_se_reshape[0][0]']	221280	
block6b_se_expand (Conv2D) (None, 1, 1, 2304) ['block6b_se_reduce[0][0]']	223488	
block6b_se_excite (Multipl (None, 4, 4, 2304)	0	

```

['block6b_activation[0][0]',
 y)
'block6b_se_expand[0][0]']

block6b_project_conv (Conv (None, 4, 4, 384) 884736
['block6b_se_excite[0][0]']
2D)

block6b_project_bn (BatchN (None, 4, 4, 384) 1536
['block6b_project_conv[0][0]']
ormalization)

block6b_drop (Dropout) (None, 4, 4, 384) 0
['block6b_project_bn[0][0]']

block6b_add (Add) (None, 4, 4, 384) 0
['block6b_drop[0][0]',
'block6a_project_bn[0][0]']

block6c_expand_conv (Conv2 (None, 4, 4, 2304) 884736
['block6b_add[0][0]']
D)

block6c_expand_bn (BatchNo (None, 4, 4, 2304) 9216
['block6c_expand_conv[0][0]']
rmalization)

block6c_expand_activation (None, 4, 4, 2304) 0
['block6c_expand_bn[0][0]']
(Activation)

block6c_dwconv (DepthwiseC (None, 4, 4, 2304) 57600
['block6c_expand_activation[0]
onv2D) [0]']

block6c_bn (BatchNormaliza (None, 4, 4, 2304) 9216
['block6c_dwconv[0][0]']
tion)

block6c_activation (Activa (None, 4, 4, 2304) 0
['block6c_bn[0][0]']
tion)

block6c_se_squeeze (Global (None, 2304) 0
['block6c_activation[0][0]']
AveragePooling2D)

block6c_se_reshape (Reshap (None, 1, 1, 2304) 0

```

['block6c_se_squeeze[0][0]'] e)		
block6c_se_reduce (Conv2D) ['block6c_se_reshape[0][0]']	(None, 1, 1, 96)	221280
block6c_se_expand (Conv2D) ['block6c_se_reduce[0][0]']	(None, 1, 1, 2304)	223488
block6c_se_excite (Multipl ['block6c_activation[0][0]', y) 'block6c_se_expand[0][0]']	(None, 4, 4, 2304)	0
block6c_project_conv (Conv ['block6c_se_excite[0][0]'] 2D)	(None, 4, 4, 384)	884736
block6c_project_bn (BatchN ['block6c_project_conv[0][0]'] ormalization)	(None, 4, 4, 384)	1536
block6c_drop (Dropout) ['block6c_project_bn[0][0]']	(None, 4, 4, 384)	0
block6c_add (Add) ['block6c_drop[0][0]', 'block6b_add[0][0]']	(None, 4, 4, 384)	0
block6d_expand_conv (Conv2 ['block6c_add[0][0]'] D)	(None, 4, 4, 2304)	884736
block6d_expand_bn (BatchNo ['block6d_expand_conv[0][0]'] rmalization)	(None, 4, 4, 2304)	9216
block6d_expand_activation ['block6d_expand_bn[0][0]'] (Activation)	(None, 4, 4, 2304)	0
block6d_dwconv (DepthwiseC ['block6d_expand_activation[0] onv2D)	(None, 4, 4, 2304)	57600
block6d_bn (BatchNormaliza ['block6d_dwconv[0][0]'] tion)	(None, 4, 4, 2304)	9216

[0]']

block6d_activation (Activation) ['block6d_bn[0][0]']	(None, 4, 4, 2304)	0
block6d_se_squeeze (Global AveragePooling2D) ['block6d_activation[0][0]']	(None, 2304)	0
block6d_se_reshape (Reshape) ['block6d_se_squeeze[0][0]']	(None, 1, 1, 2304)	0
block6d_se_reduce (Conv2D) ['block6d_se_reshape[0][0]']	(None, 1, 1, 96)	221280
block6d_se_expand (Conv2D) ['block6d_se_reduce[0][0]']	(None, 1, 1, 2304)	223488
block6d_se_excite (Multiply) ['block6d_activation[0][0]', 'block6d_se_expand[0][0]']	(None, 4, 4, 2304)	0
block6d_project_conv (Conv2D) ['block6d_se_excite[0][0]']	(None, 4, 4, 384)	884736
block6d_project_bn (Batch Normalization) ['block6d_project_conv[0][0]']	(None, 4, 4, 384)	1536
block6d_drop (Dropout) ['block6d_project_bn[0][0]']	(None, 4, 4, 384)	0
block6d_add (Add) ['block6d_drop[0][0]', 'block6c_add[0][0]']	(None, 4, 4, 384)	0
block6e_expand_conv (Conv2D) ['block6d_add[0][0]']	(None, 4, 4, 2304)	884736
block6e_expand_bn (Batch Normalization) ['block6e_expand_conv[0][0]']	(None, 4, 4, 2304)	9216
block6e_expand_activation (Activation)	(None, 4, 4, 2304)	0

['block6e_expand_bn[0][0]'] (Activation)		
block6e_dwconv (DepthwiseC (None, 4, 4, 2304) ['block6e_expand_activation[0] onv2D)	57600	[0]']
block6e_bn (BatchNormaliza (None, 4, 4, 2304) ['block6e_dwconv[0][0]'] tion)	9216	
block6e_activation (Activa (None, 4, 4, 2304) ['block6e_bn[0][0]'] tion)	0	
block6e_se_squeeze (Global (None, 2304) ['block6e_activation[0][0]'] AveragePooling2D)	0	
block6e_se_reshape (Reshap (None, 1, 1, 2304) ['block6e_se_squeeze[0][0]'] e)	0	
block6e_se_reduce (Conv2D) (None, 1, 1, 96) ['block6e_se_reshape[0][0]']	221280	
block6e_se_expand (Conv2D) (None, 1, 1, 2304) ['block6e_se_reduce[0][0]']	223488	
block6e_se_excite (Multipl (None, 4, 4, 2304) ['block6e_activation[0][0]', y) 'block6e_se_expand[0][0]']	0	
block6e_project_conv (Conv (None, 4, 4, 384) ['block6e_se_excite[0][0]'] 2D)	884736	
block6e_project_bn (BatchN (None, 4, 4, 384) ['block6e_project_conv[0][0]'] ormalization)	1536	
block6e_drop (Dropout) (None, 4, 4, 384) ['block6e_project_bn[0][0]']	0	
block6e_add (Add) (None, 4, 4, 384) ['block6e_drop[0][0]', 'block6d_add[0][0]']	0	

block6f_expand_conv (Conv2 (None, 4, 4, 2304) ['block6e_add[0][0]'] D)	884736	
block6f_expand_bn (BatchNo (None, 4, 4, 2304) ['block6f_expand_conv[0][0]'] rmalization)	9216	
block6f_expand_activation (None, 4, 4, 2304) ['block6f_expand_bn[0][0]'] (Activation)	0	
block6f_dwconv (DepthwiseC (None, 4, 4, 2304) ['block6f_expand_activation[0] onv2D)	57600	[0]']
block6f_bn (BatchNormaliza (None, 4, 4, 2304) ['block6f_dwconv[0][0]'] tion)	9216	
block6f_activation (Activa (None, 4, 4, 2304) ['block6f_bn[0][0]'] tion)	0	
block6f_se_squeeze (Global (None, 2304) ['block6f_activation[0][0]'] AveragePooling2D)	0	
block6f_se_reshape (Reshap (None, 1, 1, 2304) ['block6f_se_squeeze[0][0]'] e)	0	
block6f_se_reduce (Conv2D) (None, 1, 1, 96) ['block6f_se_reshape[0][0]']	221280	
block6f_se_expand (Conv2D) (None, 1, 1, 2304) ['block6f_se_reduce[0][0]']	223488	
block6f_se_excite (Multipl (None, 4, 4, 2304) ['block6f_activation[0][0]', y) 'block6f_se_expand[0][0]']	0	
block6f_project_conv (Conv (None, 4, 4, 384) ['block6f_se_excite[0][0]'] 2D)	884736	

block6f_project_bn (BatchNormaliza ['block6f_project_conv[0][0]'] ormalization)	(None, 4, 4, 384)	1536	
block6f_drop (Dropout) ['block6f_project_bn[0][0]']	(None, 4, 4, 384)	0	
block6f_add (Add) ['block6f_drop[0][0]', 'block6e_add[0][0]']	(None, 4, 4, 384)	0	
block6g_expand_conv (Conv2D) ['block6f_add[0][0]']	(None, 4, 4, 2304)	884736	
block6g_expand_bn (BatchNormaliza ['block6g_expand_conv[0][0]'] rmalization)	(None, 4, 4, 2304)	9216	
block6g_expand_activation ['block6g_expand_bn[0][0]'] (Activation)	(None, 4, 4, 2304)	0	
block6g_dwconv (DepthwiseConv2D) ['block6g_expand_activation[0] onv2D)']	(None, 4, 4, 2304)	57600	[0]']
block6g_bn (BatchNormalization) ['block6g_dwconv[0][0]']	(None, 4, 4, 2304)	9216	
block6g_activation (Activation) ['block6g_bn[0][0]']	(None, 4, 4, 2304)	0	
block6g_se_squeeze (GlobalAveragePooling2D) ['block6g_activation[0][0]']	(None, 2304)	0	
block6g_se_reshape (Reshape) ['block6g_se_squeeze[0][0]'] e)	(None, 1, 1, 2304)	0	
block6g_se_reduce (Conv2D) ['block6g_se_reshape[0][0]']	(None, 1, 1, 96)	221280	
block6g_se_expand (Conv2D) ['block6g_se_reduce[0][0]']	(None, 1, 1, 2304)	223488	

block6g_se_excite (Multipl ['block6g_activation[0][0]', y) 'block6g_se_expand[0][0]')	(None, 4, 4, 2304)	0	
block6g_project_conv (Conv ['block6g_se_excite[0][0]'] 2D)	(None, 4, 4, 384)	884736	
block6g_project_bn (BatchN ['block6g_project_conv[0][0]'] ormalization)	(None, 4, 4, 384)	1536	
block6g_drop (Dropout) ['block6g_project_bn[0][0]']	(None, 4, 4, 384)	0	
block6g_add (Add) ['block6g_drop[0][0]', 'block6f_add[0][0]']	(None, 4, 4, 384)	0	
block6h_expand_conv (Conv2 ['block6g_add[0][0]'] D)	(None, 4, 4, 2304)	884736	
block6h_expand_bn (BatchNo ['block6h_expand_conv[0][0]'] rmalization)	(None, 4, 4, 2304)	9216	
block6h_expand_activation ['block6h_expand_bn[0][0]'] (Activation)	(None, 4, 4, 2304)	0	
block6h_dwconv (DepthwiseC ['block6h_expand_activation[0] onv2D)	(None, 4, 4, 2304)	57600	[0]']
block6h_bn (BatchNormaliza ['block6h_dwconv[0][0]'] tion)	(None, 4, 4, 2304)	9216	
block6h_activation (Activa ['block6h_bn[0][0]'] tion)	(None, 4, 4, 2304)	0	
block6h_se_squeeze (Global ['block6h_activation[0][0]'] AveragePooling2D)	(None, 2304)	0	

block6h_se_reshape (Reshape (None, 1, 1, 2304) ['block6h_se_squeeze[0][0]'] e)	0
block6h_se_reduce (Conv2D) (None, 1, 1, 96) ['block6h_se_reshape[0][0]']	221280
block6h_se_expand (Conv2D) (None, 1, 1, 2304) ['block6h_se_reduce[0][0]']	223488
block6h_se_excite (Multipl (None, 4, 4, 2304) ['block6h_activation[0][0]', y) 'block6h_se_expand[0][0]']	0
block6h_project_conv (Conv (None, 4, 4, 384) ['block6h_se_excite[0][0]'] 2D)	884736
block6h_project_bn (BatchN (None, 4, 4, 384) ['block6h_project_conv[0][0]'] ormalization)	1536
block6h_drop (Dropout) (None, 4, 4, 384) ['block6h_project_bn[0][0]']	0
block6h_add (Add) (None, 4, 4, 384) ['block6h_drop[0][0]', 'block6g_add[0][0]']	0
block6i_expand_conv (Conv2 (None, 4, 4, 2304) ['block6h_add[0][0]'] D)	884736
block6i_expand_bn (BatchNo (None, 4, 4, 2304) ['block6i_expand_conv[0][0]'] rmalization)	9216
block6i_expand_activation (None, 4, 4, 2304) ['block6i_expand_bn[0][0]'] (Activation)	0
block6i_dwconv (DepthwiseC (None, 4, 4, 2304) ['block6i_expand_activation[0] onv2D)	57600 [0]']
block6i_bn (BatchNormaliza (None, 4, 4, 2304)	9216

['block6i_dwconv[0][0]'] tion)		
block6i_activation (Activa (None, 4, 4, 2304) ['block6i_bn[0][0]'] tion)		0
block6i_se_squeeze (Global (None, 2304) ['block6i_activation[0][0]'] AveragePooling2D)		0
block6i_se_reshape (Reshap (None, 1, 1, 2304) ['block6i_se_squeeze[0][0]'] e)		0
block6i_se_reduce (Conv2D) (None, 1, 1, 96) ['block6i_se_reshape[0][0]']		221280
block6i_se_expand (Conv2D) (None, 1, 1, 2304) ['block6i_se_reduce[0][0]']		223488
block6i_se_excite (Multipl (None, 4, 4, 2304) ['block6i_activation[0][0]', y) 'block6i_se_expand[0][0]']		0
block6i_project_conv (Conv (None, 4, 4, 384) ['block6i_se_excite[0][0]'] 2D)		884736
block6i_project_bn (BatchN (None, 4, 4, 384) ['block6i_project_conv[0][0]'] ormalization)		1536
block6i_drop (Dropout) (None, 4, 4, 384) ['block6i_project_bn[0][0]']		0
block6i_add (Add) (None, 4, 4, 384) ['block6i_drop[0][0]', 'block6h_add[0][0]']		0
block6j_expand_conv (Conv2 (None, 4, 4, 2304) ['block6i_add[0][0]'] D)		884736
block6j_expand_bn (BatchNo (None, 4, 4, 2304) ['block6j_expand_conv[0][0]'] rmalization)		9216

block6j_expand_activation (None, 4, 4, 2304) ['block6j_expand_bn[0][0]'] (Activation)	0	
block6j_dwconv (DepthwiseC (None, 4, 4, 2304) ['block6j_expand_activation[0] onv2D)	57600	[0]']
block6j_bn (BatchNormaliza (None, 4, 4, 2304) ['block6j_dwconv[0][0]'] tion)	9216	
block6j_activation (Activa (None, 4, 4, 2304) ['block6j_bn[0][0]'] tion)	0	
block6j_se_squeeze (Global (None, 2304) ['block6j_activation[0][0]'] AveragePooling2D)	0	
block6j_se_reshape (Reshap (None, 1, 1, 2304) ['block6j_se_squeeze[0][0]'] e)	0	
block6j_se_reduce (Conv2D) (None, 1, 1, 96) ['block6j_se_reshape[0][0]']	221280	
block6j_se_expand (Conv2D) (None, 1, 1, 2304) ['block6j_se_reduce[0][0]']	223488	
block6j_se_excite (Multipl (None, 4, 4, 2304) ['block6j_activation[0][0]', y) 'block6j_se_expand[0][0]']	0	
block6j_project_conv (Conv (None, 4, 4, 384) ['block6j_se_excite[0][0]'] 2D)	884736	
block6j_project_bn (BatchN (None, 4, 4, 384) ['block6j_project_conv[0][0]'] ormalization)	1536	
block6j_drop (Dropout) (None, 4, 4, 384) ['block6j_project_bn[0][0]']	0	
block6j_add (Add) (None, 4, 4, 384)	0	

```

['block6j_drop[0][0]',
'block6i_add[0][0]']

block6k_expand_conv (Conv2D (None, 4, 4, 2304) 884736
['block6j_add[0][0]']
D)

block6k_expand_bn (BatchNormalization (None, 4, 4, 2304) 9216
['block6k_expand_conv[0][0]']
rmalization)

block6k_expand_activation (None, 4, 4, 2304) 0
['block6k_expand_bn[0][0]']
(Activation)

block6k_dwconv (DepthwiseConv2D (None, 4, 4, 2304) 57600
['block6k_expand_activation[0]
onv2D) [0]']

block6k_bn (BatchNormalization (None, 4, 4, 2304) 9216
['block6k_dwconv[0][0]']
tion)

block6k_activation (Activation (None, 4, 4, 2304) 0
['block6k_bn[0][0]']
tion)

block6k_se_squeeze (GlobalAveragePooling2D (None, 2304) 0
['block6k_activation[0][0]']
AveragePooling2D)

block6k_se_reshape (Reshape (None, 1, 1, 2304) 0
['block6k_se_squeeze[0][0]']
e)

block6k_se_reduce (Conv2D (None, 1, 1, 96) 221280
['block6k_se_reshape[0][0]']

block6k_se_expand (Conv2D (None, 1, 1, 2304) 223488
['block6k_se_reduce[0][0]']

block6k_se_excite (Multiply (None, 4, 4, 2304) 0
['block6k_activation[0][0]',
y)
'block6k_se_expand[0][0]']

block6k_project_conv (Conv2D (None, 4, 4, 384) 884736
['block6k_se_excite[0][0]']

```

2D)		
block6k_project_bn (BatchNormaliza- tion) ['block6k_project_conv[0][0]']	(None, 4, 4, 384)	1536
block6k_drop (Dropout) ['block6k_project_bn[0][0]']	(None, 4, 4, 384)	0
block6k_add (Add) ['block6k_drop[0][0]', 'block6j_add[0][0]']	(None, 4, 4, 384)	0
block6l_expand_conv (Conv2D) ['block6k_add[0][0]']	(None, 4, 4, 2304)	884736
block6l_expand_bn (BatchNormaliza- tion) ['block6l_expand_conv[0][0]']	(None, 4, 4, 2304)	9216
block6l_expand_activation (Activation) ['block6l_expand_bn[0][0]']	(None, 4, 4, 2304)	0
block6l_dwconv (DepthwiseConv2D) ['block6l_expand_activation[0][0]']	(None, 4, 4, 2304)	57600
block6l_bn (BatchNormalization) ['block6l_dwconv[0][0]']	(None, 4, 4, 2304)	9216
block6l_activation (Activation) ['block6l_bn[0][0]']	(None, 4, 4, 2304)	0
block6l_se_squeeze (GlobalAveragePooling2D) ['block6l_activation[0][0]']	(None, 2304)	0
block6l_se_reshape (Reshape) ['block6l_se_squeeze[0][0]']	(None, 1, 1, 2304)	0
block6l_se_reduce (Conv2D) ['block6l_se_reshape[0][0]']	(None, 1, 1, 96)	221280

block6l_se_expand (Conv2D) ['block6l_se_reduce[0][0]']	(None, 1, 1, 2304)	223488	
block6l_se_excite (Multipl ['block6l_activation[0][0]', y) 'block6l_se_expand[0][0]']	(None, 4, 4, 2304)	0	
block6l_project_conv (Conv ['block6l_se_excite[0][0]'] 2D)	(None, 4, 4, 384)	884736	
block6l_project_bn (BatchN ['block6l_project_conv[0][0]'] ormalization)	(None, 4, 4, 384)	1536	
block6l_drop (Dropout) ['block6l_project_bn[0][0]']	(None, 4, 4, 384)	0	
block6l_add (Add) ['block6l_drop[0][0]', 'block6k_add[0][0]']	(None, 4, 4, 384)	0	
block6m_expand_conv (Conv2 ['block6l_add[0][0]'] D)	(None, 4, 4, 2304)	884736	
block6m_expand_bn (BatchNo ['block6m_expand_conv[0][0]'] rmalization)	(None, 4, 4, 2304)	9216	
block6m_expand_activation ['block6m_expand_bn[0][0]'] (Activation)	(None, 4, 4, 2304)	0	
block6m_dwconv (DepthwiseC ['block6m_expand_activation[0] onv2D)	(None, 4, 4, 2304)	57600	[0]']
block6m_bn (BatchNormaliza ['block6m_dwconv[0][0]'] tion)	(None, 4, 4, 2304)	9216	
block6m_activation (Activa ['block6m_bn[0][0]'] tion)	(None, 4, 4, 2304)	0	
block6m_se_squeeze (Global (None, 2304)	(None, 2304)	0	

['block6m_activation[0][0]'] AveragePooling2D)		
block6m_se_reshape (Reshape (None, 1, 1, 2304) ['block6m_se_squeeze[0][0]'] e)		0
block6m_se_reduce (Conv2D) (None, 1, 1, 96) ['block6m_se_reshape[0][0]']		221280
block6m_se_expand (Conv2D) (None, 1, 1, 2304) ['block6m_se_reduce[0][0]']		223488
block6m_se_excite (Multipl (None, 4, 4, 2304) ['block6m_activation[0][0]', y) 'block6m_se_expand[0][0]']		0
block6m_project_conv (Conv (None, 4, 4, 384) ['block6m_se_excite[0][0]'] 2D)		884736
block6m_project_bn (BatchN (None, 4, 4, 384) ['block6m_project_conv[0][0]'] ormalization)		1536
block6m_drop (Dropout) (None, 4, 4, 384) ['block6m_project_bn[0][0]']		0
block6m_add (Add) (None, 4, 4, 384) ['block6m_drop[0][0]', 'block6l_add[0][0]']		0
block7a_expand_conv (Conv2 (None, 4, 4, 2304) ['block6m_add[0][0]'] D)		884736
block7a_expand_bn (BatchNo (None, 4, 4, 2304) ['block7a_expand_conv[0][0]'] rmalization)		9216
block7a_expand_activation (None, 4, 4, 2304) ['block7a_expand_bn[0][0]'] (Activation)		0
block7a_dwconv (DepthwiseC (None, 4, 4, 2304) ['block7a_expand_activation[0] onv2D)		20736

[0]']

block7a_bn (BatchNormaliza ['block7a_dwconv[0][0]'] tion)	(None, 4, 4, 2304)	9216
block7a_activation (Activa ['block7a_bn[0][0]'] tion)	(None, 4, 4, 2304)	0
block7a_se_squeeze (Global ['block7a_activation[0][0]'] AveragePooling2D)	(None, 2304)	0
block7a_se_reshape (Reshap ['block7a_se_squeeze[0][0]'] e)	(None, 1, 1, 2304)	0
block7a_se_reduce (Conv2D) ['block7a_se_reshape[0][0]']	(None, 1, 1, 96)	221280
block7a_se_expand (Conv2D) ['block7a_se_reduce[0][0]']	(None, 1, 1, 2304)	223488
block7a_se_excite (Multipl ['block7a_activation[0][0]', y] 'block7a_se_expand[0][0]']	(None, 4, 4, 2304)	0
block7a_project_conv (Conv ['block7a_se_excite[0][0]'] 2D)	(None, 4, 4, 640)	1474560
block7a_project_bn (BatchN ['block7a_project_conv[0][0]'] ormalization)	(None, 4, 4, 640)	2560
block7b_expand_conv (Conv2 ['block7a_project_bn[0][0]'] D)	(None, 4, 4, 3840)	2457600
block7b_expand_bn (BatchNo ['block7b_expand_conv[0][0]'] rmalization)	(None, 4, 4, 3840)	15360
block7b_expand_activation ['block7b_expand_bn[0][0]'] (Activation)	(None, 4, 4, 3840)	0

block7b_dwconv (DepthwiseC (None, 4, 4, 3840) ['block7b_expand_activation[0] onv2D)	34560	[0] ']
block7b_bn (BatchNormaliza (None, 4, 4, 3840) ['block7b_dwconv[0][0] '] tion)	15360	
block7b_activation (Activa (None, 4, 4, 3840) ['block7b_bn[0][0] '] tion)	0	
block7b_se_squeeze (Global (None, 3840) ['block7b_activation[0][0] '] AveragePooling2D)	0	
block7b_se_reshape (Reshap (None, 1, 1, 3840) ['block7b_se_squeeze[0][0] '] e)	0	
block7b_se_reduce (Conv2D) (None, 1, 1, 160) ['block7b_se_reshape[0][0] ']	614560	
block7b_se_expand (Conv2D) (None, 1, 1, 3840) ['block7b_se_reduce[0][0] ']	618240	
block7b_se_excite (Multipl (None, 4, 4, 3840) ['block7b_activation[0][0] ', y) 'block7b_se_expand[0][0] ']	0	
block7b_project_conv (Conv (None, 4, 4, 640) ['block7b_se_excite[0][0] '] 2D)	2457600	
block7b_project_bn (BatchN (None, 4, 4, 640) ['block7b_project_conv[0][0] '] ormalization)	2560	
block7b_drop (Dropout) (None, 4, 4, 640) ['block7b_project_bn[0][0] ']	0	
block7b_add (Add) (None, 4, 4, 640) ['block7b_drop[0][0] ', 'block7a_project_bn[0][0] ']	0	
block7c_expand_conv (Conv2 (None, 4, 4, 3840) ['block7b_add[0][0] ']	2457600	

D)

block7c_expand_bn (BatchNo (None, 4, 4, 3840) ['block7c_expand_conv[0][0]'] rmalization)	15360	
block7c_expand_activation (None, 4, 4, 3840) ['block7c_expand_bn[0][0]'] (Activation)	0	
block7c_dwconv (DepthwiseC (None, 4, 4, 3840) ['block7c_expand_activation[0] onv2D)	34560	[0]']
block7c_bn (BatchNormaliza (None, 4, 4, 3840) ['block7c_dwconv[0][0]'] tion)	15360	
block7c_activation (Activa (None, 4, 4, 3840) ['block7c_bn[0][0]'] tion)	0	
block7c_se_squeeze (Global (None, 3840) ['block7c_activation[0][0]'] AveragePooling2D)	0	
block7c_se_reshape (Reshap (None, 1, 1, 3840) ['block7c_se_squeeze[0][0]'] e)	0	
block7c_se_reduce (Conv2D) (None, 1, 1, 160) ['block7c_se_reshape[0][0]']	614560	
block7c_se_expand (Conv2D) (None, 1, 1, 3840) ['block7c_se_reduce[0][0]']	618240	
block7c_se_excite (Multipl (None, 4, 4, 3840) ['block7c_activation[0][0]'], y) 'block7c_se_expand[0][0]']	0	
block7c_project_conv (Conv (None, 4, 4, 640) ['block7c_se_excite[0][0]'] 2D)	2457600	
block7c_project_bn (BatchN (None, 4, 4, 640) ['block7c_project_conv[0][0]'] ormalization)	2560	

block7c_drop (Dropout) ['block7c_project_bn[0][0]']	(None, 4, 4, 640)	0	
block7c_add (Add) ['block7c_drop[0][0]', 'block7b_add[0][0]']	(None, 4, 4, 640)	0	
block7d_expand_conv (Conv2D) ['block7c_add[0][0]']	(None, 4, 4, 3840)	2457600	
block7d_expand_bn (BatchNormalization) ['block7d_expand_conv[0][0]']	(None, 4, 4, 3840)	15360	
block7d_expand_activation (Activation) ['block7d_expand_bn[0][0]']	(None, 4, 4, 3840)	0	
block7d_dwconv (DepthwiseConv2D) ['block7d_expand_activation[0][0]']	(None, 4, 4, 3840)	34560	[0]']
block7d_bn (BatchNormalization) ['block7d_dwconv[0][0]']	(None, 4, 4, 3840)	15360	
block7d_activation (Activation) ['block7d_bn[0][0]']	(None, 4, 4, 3840)	0	
block7d_se_squeeze (GlobalAveragePooling2D) ['block7d_activation[0][0]']	(None, 3840)	0	
block7d_se_reshape (Reshape) ['block7d_se_squeeze[0][0]']	(None, 1, 1, 3840)	0	
block7d_se_reduce (Conv2D) ['block7d_se_reshape[0][0]']	(None, 1, 1, 160)	614560	
block7d_se_expand (Conv2D) ['block7d_se_reduce[0][0]']	(None, 1, 1, 3840)	618240	
block7d_se_excite (Multiply) ['block7d_activation[0][0]',	(None, 4, 4, 3840)	0	

```

y)
'block7d_se_expand[0][0]']

block7d_project_conv (Conv (None, 4, 4, 640) 2457600
['block7d_se_excite[0][0]']
2D)

block7d_project_bn (BatchN (None, 4, 4, 640) 2560
['block7d_project_conv[0][0]']
ormalization)

block7d_drop (Dropout) (None, 4, 4, 640) 0
['block7d_project_bn[0][0]']

block7d_add (Add) (None, 4, 4, 640) 0
['block7d_drop[0][0]',
'block7c_add[0][0]']

top_conv (Conv2D) (None, 4, 4, 2560) 1638400
['block7d_add[0][0]']

top_bn (BatchNormalization (None, 4, 4, 2560) 10240
['top_conv[0][0]']
)

top_activation (Activation (None, 4, 4, 2560) 0
['top_bn[0][0]']
)

conv2d_transpose_9 (Conv2D (None, 8, 8, 960) 2211936
['top_activation[0][0]']
Transpose)
0

concatenate_8 (Concatenate (None, 8, 8, 1920) 0
['conv2d_transpose_9[0][0]',
)
'block5a_activation[0][0]']

dropout_12 (Dropout) (None, 8, 8, 1920) 0
['concatenate_8[0][0]']

conv2d_28 (Conv2D) (None, 8, 8, 960) 1658976
['dropout_12[0][0]']
0

batch_normalization_26 (Ba (None, 8, 8, 960) 3840
['conv2d_28[0][0]']
tchNormalization)

```

activation_26 (Activation) (None, 8, 8, 960)	0
['batch_normalization_26[0][0]	']
conv2d_29 (Conv2D) (None, 8, 8, 960)	8295360
['activation_26[0][0]']	
batch_normalization_27 (Batch Normalization) (None, 8, 8, 960)	3840
['conv2d_29[0][0]']	
tchNormalization)	
activation_27 (Activation) (None, 8, 8, 960)	0
['batch_normalization_27[0][0]	']
conv2d_transpose_10 (Conv2DTranspose) (None, 16, 16, 288)	2488608
['activation_27[0][0]']	
DTranspose)	
concatenate_9 (Concatenate) (None, 16, 16, 576)	0
['conv2d_transpose_10[0][0]',	
)	
'block3a_activation[0][0]']	
dropout_13 (Dropout) (None, 16, 16, 576)	0
['concatenate_9[0][0]']	
conv2d_30 (Conv2D) (None, 16, 16, 288)	1493280
['dropout_13[0][0]']	
batch_normalization_28 (Batch Normalization) (None, 16, 16, 288)	1152
['conv2d_30[0][0]']	
tchNormalization)	
activation_28 (Activation) (None, 16, 16, 288)	0
['batch_normalization_28[0][0]	']
conv2d_31 (Conv2D) (None, 16, 16, 288)	746784
['activation_28[0][0]']	
batch_normalization_29 (Batch Normalization) (None, 16, 16, 288)	1152
['conv2d_31[0][0]']	
tchNormalization)	
activation_29 (Activation) (None, 16, 16, 288)	0
['batch_normalization_29[0][0]	

```

    conv2d_transpose_11 (Conv2 (None, 32, 32, 192) 497856
['activation_29[0][0]']
DTranspose)

    concatenate_10 (Concatenat (None, 32, 32, 384) 0
['conv2d_transpose_11[0][0]',
e)
'block2a_activation[0][0]']

    dropout_14 (Dropout) (None, 32, 32, 384) 0
['concatenate_10[0][0]']

    conv2d_32 (Conv2D) (None, 32, 32, 192) 663744
['dropout_14[0][0]']

    batch_normalization_30 (Ba (None, 32, 32, 192) 768
['conv2d_32[0][0]']
tchNormalization)

    activation_30 (Activation) (None, 32, 32, 192) 0
['batch_normalization_30[0][0]

    conv2d_33 (Conv2D) (None, 32, 32, 192) 331968
['activation_30[0][0]']

    batch_normalization_31 (Ba (None, 32, 32, 192) 768
['conv2d_33[0][0]']
tchNormalization)

    activation_31 (Activation) (None, 32, 32, 192) 0
['batch_normalization_31[0][0]

    conv2d_transpose_12 (Conv2 (None, 64, 64, 64) 110656
['activation_31[0][0]']
DTranspose)

    concatenate_11 (Concatenat (None, 64, 64, 128) 0
['conv2d_transpose_12[0][0]',
e)
'block1a_activation[0][0]']

    dropout_15 (Dropout) (None, 64, 64, 128) 0
['concatenate_11[0][0]']

```

```

conv2d_34 (Conv2D)          (None, 64, 64, 64)          73792
['dropout_15[0][0]']

batch_normalization_32 (Ba (None, 64, 64, 64)          256
['conv2d_34[0][0]']
tchNormalization)

activation_32 (Activation)  (None, 64, 64, 64)          0
['batch_normalization_32[0][0]

conv2d_35 (Conv2D)          (None, 64, 64, 64)          36928
['activation_32[0][0]']

batch_normalization_33 (Ba (None, 64, 64, 64)          256
['conv2d_35[0][0]']
tchNormalization)

activation_33 (Activation)  (None, 64, 64, 64)          0
['batch_normalization_33[0][0]

conv2d_transpose_13 (Conv2 (None, 128, 128, 32)          18464
['activation_33[0][0]']
DTranspose)

conv2d_36 (Conv2D)          (None, 128, 128, 2)          66
['conv2d_transpose_13[0][0]']

```

```

=====
=====
Total params: 117576345 (448.52 MB)
Trainable params: 117259602 (447.31 MB)
Non-trainable params: 316743 (1.21 MB)
-----
-----

```

9.2 Task 2.2 Train the transfer learning model and plot the training history

Feel free to use the `plot_training_history` function from the provided library `utilities.py`

9.2.1 Training

```
[20]: early_stopping = ks.callbacks.EarlyStopping(monitor='val_F1_score',
                                                    patience=5,
                                                    mode='max',
                                                    restore_best_weights=True)
```

```

# Compile the model
model_pre.compile(optimizer=ks.optimizers.Adam(learning_rate=1e-4),
                  loss='categorical_crossentropy',
                  metrics=[FalseNegatives(),
                           FalsePositives(),
                           TrueNegatives(),
                           TruePositives(),
                           F1_score])

# Train the model
model_pre_history = model_pre.fit(X_train, y_train,
                                  epochs=3, # more on orion
                                  batch_size=32,
                                  validation_split=(1/8),
                                  callbacks=[early_stopping])

model_pre.save('model_pre.keras')

```

```

Epoch 1/3
197/197 [=====] - 1114s 5s/step - loss: 0.3740 -
false_negatives_3: 26558032.0000 - false_positives_3: 10120247.0000 -
true_negatives_3: 76661208.0000 - true_positives_3: 93098960.0000 - F1_score:
0.8162 - val_loss: 0.8564 - val_false_negatives_3: 9140217.0000 -
val_false_positives_3: 6900541.0000 - val_true_negatives_3: 5605383.0000 -
val_true_positives_3: 7845059.0000 - val_F1_score: 0.4294
Epoch 2/3
197/197 [=====] - 984s 5s/step - loss: 0.3312 -
false_negatives_3: 22397440.0000 - false_positives_3: 5959666.0000 -
true_negatives_3: 80821728.0000 - true_positives_3: 97259520.0000 - F1_score:
0.8580 - val_loss: 0.9243 - val_false_negatives_3: 6288623.0000 -
val_false_positives_3: 4048944.0000 - val_true_negatives_3: 8456980.0000 -
val_true_positives_3: 10696653.0000 - val_F1_score: 0.6389
Epoch 3/3
197/197 [=====] - 979s 5s/step - loss: 0.3217 -
false_negatives_3: 22307842.0000 - false_positives_3: 5870070.0000 -
true_negatives_3: 80911320.0000 - true_positives_3: 97349160.0000 - F1_score:
0.8589 - val_loss: 0.7280 - val_false_negatives_3: 4825128.0000 -
val_false_positives_3: 2585449.0000 - val_true_negatives_3: 9920475.0000 -
val_true_positives_3: 12160148.0000 - val_F1_score: 0.7529

```

9.2.2 Plotting

```

[23]: rawDF = pd.DataFrame(model_pre_history.history)
      plotDF = pd.DataFrame()

      try:
          # Find the number behind the _ in the keys

```



```

    number = rawDF.columns[1].split('_')[-1]
    rawDF = rawDF.rename(columns={'true_positives_' + number :
↪ 'true_positives', 'true_negatives_' + number: 'true_negatives',
↪ 'false_positives_' + number: 'false_positives', 'false_negatives_' + number:
↪ 'false_negatives',
                                'val_true_positives_' + number:
↪ 'val_true_positives', 'val_true_negatives_' + number: 'val_true_negatives',
↪ 'val_false_positives_' + number: 'val_false_positives',
↪ 'val_false_negatives_' + number: 'val_false_negatives'})
except:
    pass

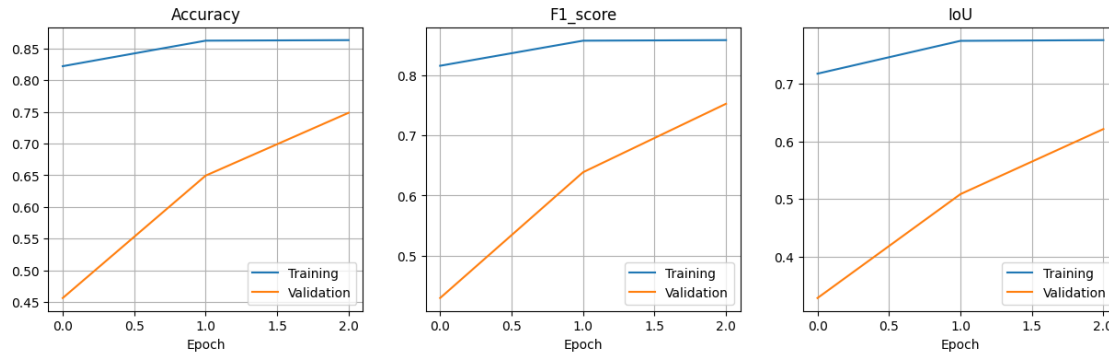
plotDF['Accuracy']      = (rawDF['true_positives'] + rawDF['true_negatives']) /
↪ (rawDF['true_positives'] + rawDF['true_negatives'] +
↪ rawDF['false_positives'] + rawDF['false_negatives'])
plotDF['val_Accuracy'] = (rawDF['val_true_positives'] +
↪ rawDF['val_true_negatives']) / (rawDF['val_true_positives'] +
↪ rawDF['val_true_negatives'] + rawDF['val_false_positives'] +
↪ rawDF['val_false_negatives'])

plotDF['IoU']           = rawDF['true_positives'] / (rawDF['true_positives'] +
↪ rawDF['false_positives'] + rawDF['false_negatives'])
plotDF['val_IoU']       = rawDF['val_true_positives'] /
↪ (rawDF['val_true_positives'] + rawDF['val_false_positives'] +
↪ rawDF['val_false_negatives'])

plotDF['F1_score']      = rawDF['F1_score']
plotDF['val_F1_score'] = rawDF['val_F1_score']

list_of_metrics=['Accuracy', 'F1_score', 'IoU']
train_keys = list_of_metrics
valid_keys = ['val_' + key for key in list_of_metrics]
nr_plots = len(list_of_metrics)
fig, ax = plt.subplots(1,nr_plots,figsize=(5*nr_plots,4))
for i in range(len(list_of_metrics)):
    ax[i].plot(np.array(plotDF[train_keys[i]]), label='Training')
    ax[i].plot(np.array(plotDF[valid_keys[i]]), label='Validation')
    ax[i].set_xlabel('Epoch')
    ax[i].set_title(list_of_metrics[i])
    ax[i].grid('on')
    ax[i].legend()
fig.tight_layout
plt.show()

```



10 Part 3: Training your model Orion

Use the lecture slides from the Orion-lecture to get started. 1. Put one of your model implementations into a python script (.py) 2. Transfer that script to Orion. 3. Change the relevant path variables in your python script (path-to-data for example), and make sure that you record the time it takes to train the model in the script. This can be done using the `time` library for example. 4. Set up a SLURM-script to train your model, please use the example from the Orion lecture as a base. 5. Submit your SLURM job, and let the magic happen.

If you wish to use a model trained on Orion to make a Kaggle submission, remember to save the model, such that you can transfer it to your local computer to make a prediction on `X_test`, or test the model on Orion directly if you want to.

10.1 Tips

If you compiled, trained and stored a model on Orion with a custom performance metric (such as F1-score), remember to specify that metric when loading the model on your computer again.

Loading a saved model:

```
trained_model = tf.keras.models.load_model('some/path/to/my_trained_model.keras', custom_object_
```

Loading a checkpoint:

```
trained_model = tf.keras.saving.load_model('some/path/to/my_trained_model_checkpoint', custom_
```

11 Discussion

Question 1: Which model architectures did you explore, and what type of hyperparameter optimization did you try?

Answer 1: We experimented with the efficientnet model.

For the native u-net model we tried different multiplums of two for the number of filters in each convolutional blocks. We also experimented with different numbers of epochs and batch sizes, and how much dropout to include. We tuned the paramneters by nested for loops, just like a grid search, we should in hindsight maybe used random search to test more hyperparms. We found that the best

params were a learning rate of 0.1 , dropout of 0.2, batch size of 32, and base for the convolutional filters of 32. We also tried some augnmentation of the images with, rotating, and scaling but this did not increase the performance so was not included here. Different loss functions such as binary focal cross entropy and dice loss was also explored because of the slight imbalance in the dataset, but did not lead to any noticable increase in performance. Training without batchnormalization was also explored, but resulted in poorer performance. L2-regularization was also explored a bit, but resulted in very noisy training performance with values larger than 0.0001. We are unsure wheter l2 regularization with smaller values would be beneficial.

Question 2: Which of the model(s) did you choose to train on Orion, and how long did it take to train it on Orion?

Answer 2: We tried to train both models on Orion but got the best performance from the model native Unet model without the pretrained backbone. On the native U-net model the model used 123 seconds on google colab, and only 7 seconds per epoch on orion. Orion is 17,5 times faster than training on colab.

Question 3: What where the biggest challenges with this assignment?

Answer 3: One of the biggest challenges was implementing the model with a pretrained backbone, it was challenging to find out exactly where the skip connections should be. Implementing the U-Net model in itself was also quite hard, since we did not really know what activation function we were to use on the last layer, we tried with both sigmoid and softmax, but found that sigmoid gave the best result. Implementing data augmentation was also a bit challenging, because it was necessary to apply the same augmentation to both images and masks. Also figuring out what type, if any, of data augmentation would be useful also took some time, and we concluded that we were not able to increase the model performance by augmenting the data.

12 Kaggle submission

Evaluate your best model on the test dataset and submit your prediction to the Kaggle leaderboard. Link to the Kaggle leaderboard will be posted in the Canvas assignment.

12.0.1 Kaggle submission

```
[25]: FILE_PATH = "tree_test.h5" # If data is in same directory as Jupyter File

with h5py.File(FILE_PATH, 'r') as f:
    print('Datasets in file:', list(f.keys()))
    X_test = np.asarray(f['X'])
    print('Nr. train images: %i'%(X_test.shape[0]))

# Turn to grey scale images
X_test = np.expand_dims(X_test, -1)
X_test = X_test.astype("float32")/255
```

Datasets in file: ['X']

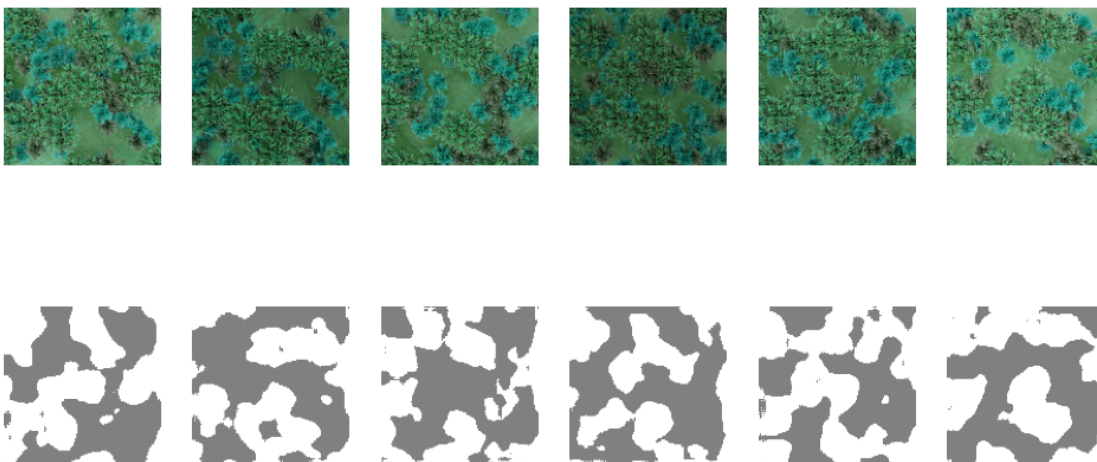
Nr. train images: 800

```
[26]: # Threshold for sigmoid
USER_DETERMINED_THRESHOLD = 0.5

y_pred      = model_pre.predict(X_test)  # Make prediction using the trained
      ↪ model
flat_y_pred = y_pred.flatten()           # Flatten prediction
flat_y_pred[flat_y_pred >= USER_DETERMINED_THRESHOLD] = 1  # Binarize
      ↪ prediction (Optional, depends on output activation used)
flat_y_pred[flat_y_pred != 1] = 0        # Binarize
      ↪ prediction (Optional, depends on output activation used)
submissionDF = pd.DataFrame()
submissionDF['ID'] = range(len(flat_y_pred))  # The submission csv
      ↪ file must have a column called 'ID'
submissionDF['Prediction'] = flat_y_pred
submissionDF.to_csv('submission.csv', index=False) # Remember to store the
      ↪ dataframe to csv without the nameless index column.
```

25/25 [=====] - 34s 951ms/step

```
[29]: # Make shure predictions look ok
plt.figure(figsize=(12, 6))
for i in range(6):
    image_x = np.squeeze(X_test[i])
    plt.subplot(2, 6, i + 1)
    plt.imshow(image_x)
    plt.axis("off")
    plt.subplot(2, 6, i + 7)
    plt.imshow(np.argmax(y_pred[i], axis=-1), cmap='gray', alpha=0.5)
    plt.axis("off")
plt.savefig('bard_2_test.png')
```



13 Orion

Here is some plots gotten from the training of the model on Orion.

13.0.1 Training history

14 Appendix

Links to fun and interesting learning resources

(not so much for this compulsory assignment, more for your own learning and interests):

- [Streamlit](#)
- [LLM-datasets](#)
- [Hands-On Graph Neural Networks Using Python](#)
- [LLM Course](#)
- [LLM by Bycroft](#)
- [Groq](#)
- [Unstructured.io](#)
- [Google Colab TPU Notebook](#)
- [LangGraph RAG Agent Llama3 Local](#)
- [LangChain Python Docs](#)
- [Nomic Embed Text v1](#)
- [DuckDB Python API Overview](#)
- [JSON Crack Editor](#)
- [Azure Data Factory Machine Learning Service](#)
- [Azure Databricks Machine Learning](#)
- [Katana ML - Machine Learning for Business Automation](#)
- [Pydeck - High-scale spatial rendering in Python, powered by deck.gl](#)