binary segmentation_2

November 19, 2022

0.0.1 Reqirements

- keras >= 2.2.0 or tensorflow >= 1.13
- segmenation-models==1.0.*
- albumentations==0.3.0

1 Loading dataset

For this example we will use **CamVid** dataset. It is a set of: - **train** images + segmentation masks - **validation** images + segmentation masks - **test** images + segmentation masks

All images have 320 pixels height and 480 pixels width. For more inforamtion about dataset visit http://mi.eng.cam.ac.uk/research/projects/VideoRec/CamVid/.

[ ]:
```
import os
from glob import glob

import cv2
from tensorflow import keras
import numpy as np
import matplotlib.pyplot as plt
import tensorflow as tf
```

```
2022-11-18 23:29:31.440137: I tensorflow/core/platform/cpu_feature_guard.cc:193]
This TensorFlow binary is optimized with oneAPI Deep Neural Network Library
(oneDNN) to use the following CPU instructions in performance-critical
operations:  AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the appropriate
compiler flags.
2022-11-18 23:29:31.559924: E tensorflow/stream_executor/cuda/cuda_blas.cc:2981]
Unable to register cuBLAS factory: Attempting to register factory for plugin
cuBLAS when one has already been registered
2022-11-18 23:29:32.021813: W
tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load
dynamic library 'libnvinfer.so.7'; dlerror: libnvinfer.so.7: cannot open shared
object file: No such file or directory; LD_LIBRARY_PATH:
/home/nkspartan/miniconda3/envs/tf-gpu/lib/python3.10/site-
packages/cv2/../../lib64::/home/nkspartan/miniconda3/envs/tf-gpu/lib/
2022-11-18 23:29:32.021882: W
```

```
tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load
dynamic library 'libnvinfer_plugin.so.7'; dlerror: libnvinfer_plugin.so.7:
cannot open shared object file: No such file or directory; LD_LIBRARY_PATH:
/home/nkspartan/miniconda3/envs/tf-gpu/lib/python3.10/site-
packages/cv2/../../lib64::/home/nkspartan/miniconda3/envs/tf-gpu/lib/
2022-11-18 23:29:32.021890: W
tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Cannot
dlopen some TensorRT libraries. If you would like to use Nvidia GPU with
TensorRT, please make sure the missing libraries mentioned above are installed
properly.
```

[ ]:
```
gpus = tf.config.experimental.list_physical_devices('GPU')
for gpu in gpus:
  tf.config.experimental.set_memory_growth(gpu, True)
```

```
2022-11-18 23:29:32.959845: I
tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:980] successful NUMA node
read from SysFS had negative value (-1), but there must be at least one NUMA
node, so returning NUMA node zero
2022-11-18 23:29:32.964213: I
tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:980] successful NUMA node
read from SysFS had negative value (-1), but there must be at least one NUMA
node, so returning NUMA node zero
2022-11-18 23:29:32.964390: I
tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:980] successful NUMA node
read from SysFS had negative value (-1), but there must be at least one NUMA
node, so returning NUMA node zero
```

[ ]:
```
DATA_DIR = './dataset/dataset_v3'
```

[ ]:
```
x_train_dir = os.path.join(DATA_DIR, 'train/images')
y_train_dir = os.path.join(DATA_DIR, 'train/masks')

x_valid_dir = os.path.join(DATA_DIR, 'validation/images')
y_valid_dir = os.path.join(DATA_DIR, 'validation/masks')

x_test_dir = os.path.join(DATA_DIR, 'test/images')
y_test_dir = os.path.join(DATA_DIR, 'test/masks')
```

2 Dataloader and utility functions

[ ]:
```
# helper function for data visualization
def visualize(**images):
    """PLot images in one row."""
    n = len(images)
    plt.figure(figsize=(16, 5))
```

```
    for i, (name, image) in enumerate(images.items()):
        plt.subplot(1, n, i + 1)
        plt.xticks([])
        plt.yticks([])
        plt.title(' '.join(name.split('_')).title())
        plt.imshow(image)
    plt.show()

# helper function for data visualization
def denormalize(x):
    """Scale image to range 0..1 for correct plot"""
    x_max = np.percentile(x, 98)
    x_min = np.percentile(x, 2)
    x = (x - x_min) / (x_max - x_min)
    x = x.clip(0, 1)
    return x

def cartoonize_image(img):
    # Defining input data for clustering
    data = np.float32(img).reshape((-1, 3))

    # Defining criteria
    criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 20, 1.0)
    # Applying cv2.kmeans function
    _, label, center = cv2.kmeans(data, 8, None, criteria, 10, cv2.
 ↪KMEANS_RANDOM_CENTERS)

    center = np.uint8(center)
    # Reshape the output data to the size of input image
    result = center[label.flatten()]
    result = result.reshape(img.shape)

    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    # Perform adaptive threshold
    edges  = cv2.adaptiveThreshold(gray, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C,␣
 ↪cv2.THRESH_BINARY, 9, 8)

    blurred = cv2.medianBlur(result, 3)
    # Combine the result and edges to get final cartoon effect
    cartoon = cv2.bitwise_and(blurred, blurred, mask=edges)

    return cartoon

def overlay_edges(img):
    edges = cv2.Canny(img, 90, 120)    # canny edge detector

    img[edges == 255] = [255, 0, 0]  # turn edges to red
```

```
    return img

def unsharp_mask(image, kernel_size=(5, 5), sigma=1.0, amount=1.0, threshold=0):
    """Return a sharpened version of the image, using an unsharp mask."""
    blurred = cv2.GaussianBlur(image, kernel_size, sigma)
    sharpened = float(amount + 1) * image - float(amount) * blurred
    sharpened = np.maximum(sharpened, np.zeros(sharpened.shape))
    sharpened = np.minimum(sharpened, 255 * np.ones(sharpened.shape))
    sharpened = sharpened.round().astype(np.uint8)
    if threshold > 0:
        low_contrast_mask = np.absolute(image - blurred) < threshold
        np.copyto(sharpened, image, where=low_contrast_mask)
    return sharpened

def sharpen_image(image):
    kernel = np.array([[-1,-1,-1],
                       [-1, 9,-1],
                       [-1,-1,-1]])
    sharpened = cv2.filter2D(image, -1, kernel)

    return sharpened

def denoise_image(image):
    noiseless_image = cv2.bilateralFilter(image, 11, 41, 21)

    return noiseless_image



# classes for data loading and preprocessing
class Dataset:
    """CamVid Dataset. Read images, apply augmentation and preprocessing␣
 ↪transformations.

    Args:
        images_dir (str): path to images folder
        masks_dir (str): path to segmentation masks folder
        class_values (list): values of classes to extract from segmentation mask
        augmentation (albumentations.Compose): data transfromation pipeline
            (e.g. flip, scale, etc.)
        preprocessing (albumentations.Compose): data preprocessing
            (e.g. noralization, shape manipulation, etc.)

    """

    CLASSES = ['river', 'not_river']
```

```
def __init__(
        self,
        images_dir,
        masks_dir,
        classes=None,
        augmentation=None,
        preprocessing=None,
):
    self.ids = os.listdir(images_dir)
    self.images_fps = [str(i) for i in glob(images_dir + '/*')]
    self.masks_fps = [str(i) for i in glob(masks_dir + '/*')]
    self.images_fps.sort()
    self.masks_fps.sort()

    # convert str names to class values on masks
    self.class_values = classes

    self.augmentation = augmentation
    self.preprocessing = preprocessing

def __getitem__(self, i):

    # read data
    image = cv2.imread(self.images_fps[i])
    image = cv2.resize(image, (320, 320), interpolation = cv2.INTER_AREA)
    #image = cartoonize_image(image)
    #image = overlay_edges(image)
    #image = unsharp_mask(image)
    #image = sharpen_image(image)
    image = denoise_image(image)

    mask = cv2.imread(self.masks_fps[i])
    mask = cv2.cvtColor(mask, cv2.COLOR_BGR2GRAY)
    mask = cv2.resize(mask, (320, 320), interpolation = cv2.INTER_AREA)
    mask = np.expand_dims(mask, axis=-1)
    mask = mask.astype('float')

    # extract certain classes from mask (e.g. cars)
    #masks = [(mask == v) for v in self.class_values]
    #mask = np.stack(masks, axis=-1).astype('float')

    # apply augmentations
    if self.augmentation:
        sample = self.augmentation(image=image, mask=mask)
        image, mask = sample['image'], sample['mask']
```

```
        # apply preprocessing
        if self.preprocessing:
            sample = self.preprocessing(image=image, mask=mask)
            image, mask = sample['image'], sample['mask']

        return image, mask

    def __len__(self):
        return len(self.ids)


class Dataloder(keras.utils.Sequence):
    """Load data from dataset and form batches

    Args:
        dataset: instance of Dataset class for image loading and preprocessing.
        batch_size: Integet number of images in batch.
        shuffle: Boolean, if `True` shuffle image indexes each epoch.
    """

    def __init__(self, dataset, batch_size=1, shuffle=False):
        self.dataset = dataset
        self.batch_size = batch_size
        self.shuffle = shuffle
        self.indexes = np.arange(len(dataset))

        self.on_epoch_end()

    def __getitem__(self, i):

        # collect batch data
        start = i * self.batch_size
        stop = (i + 1) * self.batch_size
        data = []
        for j in range(start, stop):
            data.append(self.dataset[j])

        # transpose list of lists
        batch = [np.stack(samples, axis=0) for samples in zip(*data)]

        return batch

    def __len__(self):
        """Denotes the number of batches per epoch"""
        return len(self.indexes) // self.batch_size

    def on_epoch_end(self):
```

```
        """Callback function to shuffle indexes each epoch"""
        if self.shuffle:
            self.indexes = np.random.permutation(self.indexes)
```

[ ]:
```
# Lets look at data we have
dataset = Dataset(x_train_dir, y_train_dir, classes=[0, 1])

image, mask = dataset[5] # get some sample
print(mask.shape)
visualize(
    image=image,
    mask=mask
)
```

```
(320, 320, 1)
```

2.0.1 Augmentations

Data augmentation is a powerful technique to increase the amount of your data and prevent model overfitting.
If you not familiar with such trick read some of these articles: - The Effectiveness of Data Augmentation in Image Classification using Deep Learning - Data Augmentation | How to use Deep Learning when you have Limited Data - Data Augmentation Experimentation

Since our dataset is very small we will apply a large number of different augmentations: - horizontal flip - affine transforms - perspective transforms - brightness/contrast/colors manipulations - image bluring and sharpening - gaussian noise - random crops

All this transforms can be easily applied with **Albumentations** - fast augmentation library. For detailed explanation of image transformations you can look at kaggle salt segmentation exmaple provided by **Albumentations** authors.

[ ]:
```
import albumentations as A
```

[ ]:
```
def round_clip_0_1(x, **kwargs):
    return x.round().clip(0, 1)

# define heavy augmentations
def get_training_augmentation():
    train_transform = [

        A.HorizontalFlip(p=0.5),

        A.ShiftScaleRotate(scale_limit=0.5, rotate_limit=0, shift_limit=0.1,␣
 ↪p=1, border_mode=0),

        A.PadIfNeeded(min_height=320, min_width=320, always_apply=True,␣
 ↪border_mode=0),
        A.RandomCrop(height=320, width=320, always_apply=True),

        A.GaussNoise(p=0.2),
        A.Perspective(p=0.5),

        A.OneOf(
            [
                A.CLAHE(p=1),
                A.RandomBrightness(p=1),
                A.RandomGamma(p=1),
            ],
            p=0.9,
        ),

        A.OneOf(
            [
                A.Sharpen(p=1),
                #A.Blur(blur_limit=3, p=1),
                #A.MotionBlur(blur_limit=3, p=1),
            ],
            p=0.9,
        ),

        A.OneOf(
            [
                A.RandomContrast(p=1),
                A.HueSaturationValue(p=1),
            ],
            p=0.9,
        ),
        A.Lambda(mask=round_clip_0_1)
    ]
    return A.Compose(train_transform)
```

```
def get_validation_augmentation():
    """Add paddings to make image shape divisible by 32"""
    test_transform = [
        A.PadIfNeeded(384, 480)
    ]
    return A.Compose(test_transform)

def get_preprocessing(preprocessing_fn):
    """Construct preprocessing transform

    Args:
        preprocessing_fn (callbale): data normalization function
            (can be specific for each pretrained neural network)
    Return:
        transform: albumentations.Compose

    """

    _transform = [
        A.Lambda(image=preprocessing_fn),
    ]
    return A.Compose(_transform)
```

[ ]:
```
# Lets look at augmented data we have
dataset = Dataset(x_train_dir, y_train_dir, classes=[0, 1],␣
 ↪augmentation=get_training_augmentation())

image, mask = dataset[12] # get some sample
visualize(
    image=image,
    mask=mask
)
```

```
/home/nkspartan/miniconda3/envs/tf-gpu/lib/python3.10/site-
packages/albumentations/augmentations/transforms.py:1149: FutureWarning: This
class has been deprecated. Please use RandomBrightnessContrast
  warnings.warn(
/home/nkspartan/miniconda3/envs/tf-gpu/lib/python3.10/site-
packages/albumentations/augmentations/transforms.py:1175: FutureWarning:
RandomContrast has been deprecated. Please use RandomBrightnessContrast
  warnings.warn(
```

3 Segmentation model training

[ ]:
```
import segmentation_models as sm

# segmentation_models could also use `tf.keras` if you do not have Keras␣
↪installed
# or you could switch to other framework using `sm.set_framework('tf.keras')`
```

```
Segmentation Models: using `keras` framework.
```

[ ]:
```
BACKBONE = 'seresnext50'
BATCH_SIZE = 4
CLASSES = [1]
LR = 0.0001
EPOCHS = 70

preprocess_input = sm.get_preprocessing(BACKBONE)
```

[ ]:
```
# define network parameters
n_classes = 1  # case for binary and multiclass segmentation
activation = 'sigmoid' if n_classes == 1 else 'softmax'

#create model
model = sm.Unet(BACKBONE, classes=n_classes, activation=activation)
```

```
2022-11-18 23:29:33.788550: I tensorflow/core/platform/cpu_feature_guard.cc:193]
This TensorFlow binary is optimized with oneAPI Deep Neural Network Library
(oneDNN) to use the following CPU instructions in performance-critical
operations:  AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the appropriate
compiler flags.
```

```
2022-11-18 23:29:33.789230: I
tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:980] successful NUMA node
read from SysFS had negative value (-1), but there must be at least one NUMA
node, so returning NUMA node zero
2022-11-18 23:29:33.789467: I
tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:980] successful NUMA node
read from SysFS had negative value (-1), but there must be at least one NUMA
node, so returning NUMA node zero
2022-11-18 23:29:33.789626: I
tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:980] successful NUMA node
read from SysFS had negative value (-1), but there must be at least one NUMA
node, so returning NUMA node zero
2022-11-18 23:29:34.186928: I
tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:980] successful NUMA node
read from SysFS had negative value (-1), but there must be at least one NUMA
node, so returning NUMA node zero
2022-11-18 23:29:34.187195: I
tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:980] successful NUMA node
read from SysFS had negative value (-1), but there must be at least one NUMA
node, so returning NUMA node zero
2022-11-18 23:29:34.187516: I
tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:980] successful NUMA node
read from SysFS had negative value (-1), but there must be at least one NUMA
node, so returning NUMA node zero
2022-11-18 23:29:34.187718: I
tensorflow/core/common_runtime/gpu/gpu_device.cc:1616] Created device
/job:localhost/replica:0/task:0/device:GPU:0 with 4041 MB memory:  -> device: 0,
name: NVIDIA GeForce RTX 2060, pci bus id: 0000:08:00.0, compute capability: 7.5
```

[ ]:
```
#model.summary()
```

[ ]:
```
# define optomizer
optim = keras.optimizers.Adam(LR)

# Segmentation models losses can be combined together by '+' and scaled by␣
 ↪integer or float factor
dice_loss = sm.losses.DiceLoss()
focal_loss = sm.losses.BinaryFocalLoss() if n_classes == 1 else sm.losses.
 ↪CategoricalFocalLoss()
total_loss = dice_loss + (1 * focal_loss)

# actulally total_loss can be imported directly from library, above example␣
 ↪just show you how to manipulate with losses
# total_loss = sm.losses.binary_focal_dice_loss # or sm.losses.
 ↪categorical_focal_dice_loss

metrics = [sm.metrics.IOUScore(threshold=0.5), sm.metrics.FScore(threshold=0.5)]
```

```
# compile keras model with defined optimozer, loss and metrics
model.compile(optim, total_loss, metrics)
```

[ ]:
```
import datetime

date_actual = datetime.datetime.now().strftime("%Y%m%d-%H%M%S")
log_dir = "logs/fit/" + date_actual
tensorboard_callback = tf.keras.callbacks.TensorBoard(log_dir=log_dir,␣
↪histogram_freq=1)

es_callback = tf.keras.callbacks.EarlyStopping(monitor='val_loss', mode='min',␣
↪verbose=1, patience=16)

checkpoint_callback = tf.keras.callbacks.
↪ModelCheckpoint(filepath=f"model_weights/
↪{date_actual}_segmentation_best_weights.hdf5",
                                monitor='val_loss',
                                verbose=1,
                                save_weights_only=True,
                                save_best_only=True)
```

[ ]:
```
# Dataset for train images
train_dataset = Dataset(
    x_train_dir,
    y_train_dir,
    classes=[0, 1],
    augmentation=get_training_augmentation(),
    preprocessing=get_preprocessing(preprocess_input),
)

# Dataset for validation images
valid_dataset = Dataset(
    x_valid_dir,
    y_valid_dir,
    classes=[0, 1],
    augmentation=get_validation_augmentation(),
    preprocessing=get_preprocessing(preprocess_input),
)

train_dataloader = Dataloder(train_dataset, batch_size=BATCH_SIZE, shuffle=True)
valid_dataloader = Dataloder(valid_dataset, batch_size=1, shuffle=False)

# check shapes for errors
assert train_dataloader[0][0].shape == (BATCH_SIZE, 320, 320, 3)
assert train_dataloader[0][1].shape == (BATCH_SIZE, 320, 320, n_classes)
```

```
# define callbacks for learning rate scheduling and best checkpoints saving
callbacks = [
    checkpoint_callback,
    es_callback,
    tensorboard_callback,
    keras.callbacks.ReduceLROnPlateau(),
]
```

[ ]:
```
# train model
history = model.fit(
    train_dataloader,
    steps_per_epoch=len(train_dataloader),
    epochs=EPOCHS,
    callbacks=callbacks,
    validation_data=valid_dataloader,
    validation_steps=len(valid_dataloader),
)
```

```
Epoch 1/70

2022-11-18 23:30:02.689151: I tensorflow/stream_executor/cuda/cuda_dnn.cc:384]
Loaded cuDNN version 8100
2022-11-18 23:30:03.065232: I
tensorflow/core/platform/default/subprocess.cc:304] Start cannot spawn child
process: No such file or directory
2022-11-18 23:30:03.066223: I
tensorflow/core/platform/default/subprocess.cc:304] Start cannot spawn child
process: No such file or directory
2022-11-18 23:30:03.066235: W tensorflow/stream_executor/gpu/asm_compiler.cc:80]
Couldn't get ptxas version string: INTERNAL: Couldn't invoke ptxas --version
2022-11-18 23:30:03.067074: I
tensorflow/core/platform/default/subprocess.cc:304] Start cannot spawn child
process: No such file or directory
2022-11-18 23:30:03.067126: W
tensorflow/stream_executor/gpu/redzone_allocator.cc:314] INTERNAL: Failed to
launch ptxas
Relying on driver to perform ptx compilation.
Modify $PATH to customize ptxas location.
This message will be only logged once.
2022-11-18 23:30:03.630914: W
tensorflow/core/common_runtime/bfc_allocator.cc:290] Allocator (GPU_0_bfc) ran
out of memory trying to allocate 4.14GiB with freed_by_count=0. The caller
indicates that this is not a failure, but this may mean that there could be
performance gains if more memory were available.
2022-11-18 23:30:03.630947: W
tensorflow/core/common_runtime/bfc_allocator.cc:290] Allocator (GPU_0_bfc) ran
out of memory trying to allocate 4.14GiB with freed_by_count=0. The caller
indicates that this is not a failure, but this may mean that there could be
```

```
performance gains if more memory were available.
2022-11-18 23:30:03.665142: W
tensorflow/core/common_runtime/bfc_allocator.cc:290] Allocator (GPU_0_bfc) ran
out of memory trying to allocate 4.21GiB with freed_by_count=0. The caller
indicates that this is not a failure, but this may mean that there could be
performance gains if more memory were available.
2022-11-18 23:30:03.665172: W
tensorflow/core/common_runtime/bfc_allocator.cc:290] Allocator (GPU_0_bfc) ran
out of memory trying to allocate 4.21GiB with freed_by_count=0. The caller
indicates that this is not a failure, but this may mean that there could be
performance gains if more memory were available.
2022-11-18 23:30:03.717252: W
tensorflow/core/common_runtime/bfc_allocator.cc:290] Allocator (GPU_0_bfc) ran
out of memory trying to allocate 4.17GiB with freed_by_count=0. The caller
indicates that this is not a failure, but this may mean that there could be
performance gains if more memory were available.
2022-11-18 23:30:03.717279: W
tensorflow/core/common_runtime/bfc_allocator.cc:290] Allocator (GPU_0_bfc) ran
out of memory trying to allocate 4.17GiB with freed_by_count=0. The caller
indicates that this is not a failure, but this may mean that there could be
performance gains if more memory were available.
2022-11-18 23:30:03.746121: W
tensorflow/core/common_runtime/bfc_allocator.cc:290] Allocator (GPU_0_bfc) ran
out of memory trying to allocate 4.21GiB with freed_by_count=0. The caller
indicates that this is not a failure, but this may mean that there could be
performance gains if more memory were available.
2022-11-18 23:30:03.746146: W
tensorflow/core/common_runtime/bfc_allocator.cc:290] Allocator (GPU_0_bfc) ran
out of memory trying to allocate 4.21GiB with freed_by_count=0. The caller
indicates that this is not a failure, but this may mean that there could be
performance gains if more memory were available.
2022-11-18 23:30:03.809159: W
tensorflow/core/common_runtime/bfc_allocator.cc:290] Allocator (GPU_0_bfc) ran
out of memory trying to allocate 4.28GiB with freed_by_count=0. The caller
indicates that this is not a failure, but this may mean that there could be
performance gains if more memory were available.
2022-11-18 23:30:03.809186: W
tensorflow/core/common_runtime/bfc_allocator.cc:290] Allocator (GPU_0_bfc) ran
out of memory trying to allocate 4.28GiB with freed_by_count=0. The caller
indicates that this is not a failure, but this may mean that there could be
performance gains if more memory were available.

200/200 [==============================] - ETA: 0s - loss: 0.6454 - iou_score:
0.4822 - f1-score: 0.6281
Epoch 1: val_loss improved from inf to 0.47711, saving model to
model_weights/20221118-232939_segmentation_best_weights.hdf5

2022-11-18 23:31:24.812677: W
tensorflow/core/framework/cpu_allocator_impl.cc:82] Allocation of 1698693120
```

```
exceeds 10% of free system memory.

200/200 [==============================] - 106s 394ms/step - loss: 0.6454 -
iou_score: 0.4822 - f1-score: 0.6281 - val_loss: 0.4771 - val_iou_score: 0.6615
- val_f1-score: 0.7749 - lr: 1.0000e-04
Epoch 2/70
200/200 [==============================] - ETA: 0s - loss: 0.4660 - iou_score:
0.6325 - f1-score: 0.7610
Epoch 2: val_loss improved from 0.47711 to 0.42430, saving model to
model_weights/20221118-232939_segmentation_best_weights.hdf5

2022-11-18 23:32:40.559362: W
tensorflow/core/framework/cpu_allocator_impl.cc:82] Allocation of 1698693120
exceeds 10% of free system memory.

200/200 [==============================] - 76s 378ms/step - loss: 0.4660 -
iou_score: 0.6325 - f1-score: 0.7610 - val_loss: 0.4243 - val_iou_score: 0.7050
- val_f1-score: 0.8055 - lr: 1.0000e-04
Epoch 3/70
200/200 [==============================] - ETA: 0s - loss: 0.3642 - iou_score:
0.7076 - f1-score: 0.8191
Epoch 3: val_loss improved from 0.42430 to 0.33977, saving model to
model_weights/20221118-232939_segmentation_best_weights.hdf5

2022-11-18 23:33:55.777130: W
tensorflow/core/framework/cpu_allocator_impl.cc:82] Allocation of 1698693120
exceeds 10% of free system memory.

200/200 [==============================] - 75s 376ms/step - loss: 0.3642 -
iou_score: 0.7076 - f1-score: 0.8191 - val_loss: 0.3398 - val_iou_score: 0.7187
- val_f1-score: 0.8157 - lr: 1.0000e-04
Epoch 4/70
200/200 [==============================] - ETA: 0s - loss: 0.3174 - iou_score:
0.7374 - f1-score: 0.8394
Epoch 4: val_loss improved from 0.33977 to 0.33552, saving model to
model_weights/20221118-232939_segmentation_best_weights.hdf5

2022-11-18 23:35:11.560273: W
tensorflow/core/framework/cpu_allocator_impl.cc:82] Allocation of 1698693120
exceeds 10% of free system memory.

200/200 [==============================] - 76s 378ms/step - loss: 0.3174 -
iou_score: 0.7374 - f1-score: 0.8394 - val_loss: 0.3355 - val_iou_score: 0.7624
- val_f1-score: 0.8494 - lr: 1.0000e-04
Epoch 5/70
200/200 [==============================] - ETA: 0s - loss: 0.2699 - iou_score:
0.7657 - f1-score: 0.8595
Epoch 5: val_loss improved from 0.33552 to 0.29952, saving model to
model_weights/20221118-232939_segmentation_best_weights.hdf5

2022-11-18 23:36:26.958164: W
tensorflow/core/framework/cpu_allocator_impl.cc:82] Allocation of 1698693120
```

```
exceeds 10% of free system memory.
200/200 [==============================] - 75s 376ms/step - loss: 0.2699 -
iou_score: 0.7657 - f1-score: 0.8595 - val_loss: 0.2995 - val_iou_score: 0.7618
- val_f1-score: 0.8464 - lr: 1.0000e-04
Epoch 6/70
200/200 [==============================] - ETA: 0s - loss: 0.2155 - iou_score:
0.8129 - f1-score: 0.8918
Epoch 6: val_loss improved from 0.29952 to 0.26369, saving model to
model_weights/20221118-232939_segmentation_best_weights.hdf5
200/200 [==============================] - 76s 380ms/step - loss: 0.2155 -
iou_score: 0.8129 - f1-score: 0.8918 - val_loss: 0.2637 - val_iou_score: 0.7827
- val_f1-score: 0.8622 - lr: 1.0000e-04
Epoch 7/70
200/200 [==============================] - ETA: 0s - loss: 0.1949 - iou_score:
0.8198 - f1-score: 0.8974
Epoch 7: val_loss did not improve from 0.26369
200/200 [==============================] - 74s 368ms/step - loss: 0.1949 -
iou_score: 0.8198 - f1-score: 0.8974 - val_loss: 0.2945 - val_iou_score: 0.7283
- val_f1-score: 0.8237 - lr: 1.0000e-04
Epoch 8/70
200/200 [==============================] - ETA: 0s - loss: 0.1951 - iou_score:
0.8170 - f1-score: 0.8932
Epoch 8: val_loss improved from 0.26369 to 0.24871, saving model to
model_weights/20221118-232939_segmentation_best_weights.hdf5
200/200 [==============================] - 76s 380ms/step - loss: 0.1951 -
iou_score: 0.8170 - f1-score: 0.8932 - val_loss: 0.2487 - val_iou_score: 0.7671
- val_f1-score: 0.8506 - lr: 1.0000e-04
Epoch 9/70
200/200 [==============================] - ETA: 0s - loss: 0.1677 - iou_score:
0.8382 - f1-score: 0.9078
Epoch 9: val_loss improved from 0.24871 to 0.24012, saving model to
model_weights/20221118-232939_segmentation_best_weights.hdf5
200/200 [==============================] - 76s 379ms/step - loss: 0.1677 -
iou_score: 0.8382 - f1-score: 0.9078 - val_loss: 0.2401 - val_iou_score: 0.7727
- val_f1-score: 0.8526 - lr: 1.0000e-04
Epoch 10/70
200/200 [==============================] - ETA: 0s - loss: 0.1514 - iou_score:
0.8492 - f1-score: 0.9154
Epoch 10: val_loss improved from 0.24012 to 0.22939, saving model to
model_weights/20221118-232939_segmentation_best_weights.hdf5
200/200 [==============================] - 76s 379ms/step - loss: 0.1514 -
iou_score: 0.8492 - f1-score: 0.9154 - val_loss: 0.2294 - val_iou_score: 0.7877
- val_f1-score: 0.8653 - lr: 1.0000e-04
Epoch 11/70
200/200 [==============================] - ETA: 0s - loss: 0.1503 - iou_score:
0.8459 - f1-score: 0.9128
Epoch 11: val_loss did not improve from 0.22939
```

```
200/200 [==============================] - 74s 368ms/step - loss: 0.1503 -
iou_score: 0.8459 - f1-score: 0.9128 - val_loss: 0.2410 - val_iou_score: 0.7627
- val_f1-score: 0.8441 - lr: 1.0000e-04
Epoch 12/70
200/200 [==============================] - ETA: 0s - loss: 0.1300 - iou_score:
0.8670 - f1-score: 0.9265
Epoch 12: val_loss did not improve from 0.22939
200/200 [==============================] - 74s 369ms/step - loss: 0.1300 -
iou_score: 0.8670 - f1-score: 0.9265 - val_loss: 0.2766 - val_iou_score: 0.7697
- val_f1-score: 0.8531 - lr: 1.0000e-04
Epoch 13/70
200/200 [==============================] - ETA: 0s - loss: 0.1429 - iou_score:
0.8537 - f1-score: 0.9178
Epoch 13: val_loss did not improve from 0.22939
200/200 [==============================] - 74s 368ms/step - loss: 0.1429 -
iou_score: 0.8537 - f1-score: 0.9178 - val_loss: 0.2539 - val_iou_score: 0.7599
- val_f1-score: 0.8406 - lr: 1.0000e-04
Epoch 14/70
200/200 [==============================] - ETA: 0s - loss: 0.1310 - iou_score:
0.8652 - f1-score: 0.9250
Epoch 14: val_loss improved from 0.22939 to 0.22916, saving model to
model_weights/20221118-232939_segmentation_best_weights.hdf5
200/200 [==============================] - 76s 378ms/step - loss: 0.1310 -
iou_score: 0.8652 - f1-score: 0.9250 - val_loss: 0.2292 - val_iou_score: 0.7869
- val_f1-score: 0.8650 - lr: 1.0000e-04
Epoch 15/70
200/200 [==============================] - ETA: 0s - loss: 0.1310 - iou_score:
0.8628 - f1-score: 0.9234
Epoch 15: val_loss improved from 0.22916 to 0.22153, saving model to
model_weights/20221118-232939_segmentation_best_weights.hdf5
200/200 [==============================] - 75s 374ms/step - loss: 0.1310 -
iou_score: 0.8628 - f1-score: 0.9234 - val_loss: 0.2215 - val_iou_score: 0.7900
- val_f1-score: 0.8644 - lr: 1.0000e-04
Epoch 16/70
200/200 [==============================] - ETA: 0s - loss: 0.1098 - iou_score:
0.8823 - f1-score: 0.9356
Epoch 16: val_loss did not improve from 0.22153
200/200 [==============================] - 74s 369ms/step - loss: 0.1098 -
iou_score: 0.8823 - f1-score: 0.9356 - val_loss: 0.2295 - val_iou_score: 0.7921
- val_f1-score: 0.8667 - lr: 1.0000e-04
Epoch 17/70
200/200 [==============================] - ETA: 0s - loss: 0.1244 - iou_score:
0.8681 - f1-score: 0.9265
Epoch 17: val_loss did not improve from 0.22153
200/200 [==============================] - 74s 368ms/step - loss: 0.1244 -
iou_score: 0.8681 - f1-score: 0.9265 - val_loss: 0.2349 - val_iou_score: 0.7862
- val_f1-score: 0.8638 - lr: 1.0000e-04
Epoch 18/70
```

```
200/200 [==============================] - ETA: 0s - loss: 0.1138 - iou_score:
0.8768 - f1-score: 0.9325
Epoch 18: val_loss did not improve from 0.22153
200/200 [==============================] - 74s 370ms/step - loss: 0.1138 -
iou_score: 0.8768 - f1-score: 0.9325 - val_loss: 0.2416 - val_iou_score: 0.7652
- val_f1-score: 0.8468 - lr: 1.0000e-04
Epoch 19/70
200/200 [==============================] - ETA: 0s - loss: 0.1092 - iou_score:
0.8811 - f1-score: 0.9345
Epoch 19: val_loss did not improve from 0.22153
200/200 [==============================] - 74s 369ms/step - loss: 0.1092 -
iou_score: 0.8811 - f1-score: 0.9345 - val_loss: 0.2578 - val_iou_score: 0.7749
- val_f1-score: 0.8578 - lr: 1.0000e-04
Epoch 20/70
200/200 [==============================] - ETA: 0s - loss: 0.1041 - iou_score:
0.8879 - f1-score: 0.9386
Epoch 20: val_loss improved from 0.22153 to 0.21191, saving model to
model_weights/20221118-232939_segmentation_best_weights.hdf5
200/200 [==============================] - 76s 377ms/step - loss: 0.1041 -
iou_score: 0.8879 - f1-score: 0.9386 - val_loss: 0.2119 - val_iou_score: 0.8121
- val_f1-score: 0.8835 - lr: 1.0000e-04
Epoch 21/70
200/200 [==============================] - ETA: 0s - loss: 0.1112 - iou_score:
0.8804 - f1-score: 0.9338
Epoch 21: val_loss did not improve from 0.21191
200/200 [==============================] - 74s 369ms/step - loss: 0.1112 -
iou_score: 0.8804 - f1-score: 0.9338 - val_loss: 0.2622 - val_iou_score: 0.7934
- val_f1-score: 0.8680 - lr: 1.0000e-04
Epoch 22/70
200/200 [==============================] - ETA: 0s - loss: 0.1092 - iou_score:
0.8828 - f1-score: 0.9352
Epoch 22: val_loss did not improve from 0.21191
200/200 [==============================] - 74s 368ms/step - loss: 0.1092 -
iou_score: 0.8828 - f1-score: 0.9352 - val_loss: 0.2553 - val_iou_score: 0.7549
- val_f1-score: 0.8364 - lr: 1.0000e-04
Epoch 23/70
200/200 [==============================] - ETA: 0s - loss: 0.1182 - iou_score:
0.8707 - f1-score: 0.9280
Epoch 23: val_loss did not improve from 0.21191
200/200 [==============================] - 74s 369ms/step - loss: 0.1182 -
iou_score: 0.8707 - f1-score: 0.9280 - val_loss: 0.2124 - val_iou_score: 0.7933
- val_f1-score: 0.8717 - lr: 1.0000e-04
Epoch 24/70
200/200 [==============================] - ETA: 0s - loss: 0.0891 - iou_score:
0.9009 - f1-score: 0.9467
Epoch 24: val_loss improved from 0.21191 to 0.20038, saving model to
model_weights/20221118-232939_segmentation_best_weights.hdf5
200/200 [==============================] - 76s 379ms/step - loss: 0.0891 -
```

```
iou_score: 0.9009 - f1-score: 0.9467 - val_loss: 0.2004 - val_iou_score: 0.8017
- val_f1-score: 0.8766 - lr: 1.0000e-04
Epoch 25/70
200/200 [==============================] - ETA: 0s - loss: 0.1010 - iou_score:
0.8879 - f1-score: 0.9389
Epoch 25: val_loss did not improve from 0.20038
200/200 [==============================] - 74s 368ms/step - loss: 0.1010 -
iou_score: 0.8879 - f1-score: 0.9389 - val_loss: 0.2229 - val_iou_score: 0.7882
- val_f1-score: 0.8636 - lr: 1.0000e-04
Epoch 26/70
200/200 [==============================] - ETA: 0s - loss: 0.1072 - iou_score:
0.8821 - f1-score: 0.9349
Epoch 26: val_loss did not improve from 0.20038
200/200 [==============================] - 74s 368ms/step - loss: 0.1072 -
iou_score: 0.8821 - f1-score: 0.9349 - val_loss: 0.2271 - val_iou_score: 0.7768
- val_f1-score: 0.8574 - lr: 1.0000e-04
Epoch 27/70
200/200 [==============================] - ETA: 0s - loss: 0.0979 - iou_score:
0.8912 - f1-score: 0.9411
Epoch 27: val_loss did not improve from 0.20038
200/200 [==============================] - 74s 370ms/step - loss: 0.0979 -
iou_score: 0.8912 - f1-score: 0.9411 - val_loss: 0.2057 - val_iou_score: 0.8123
- val_f1-score: 0.8818 - lr: 1.0000e-04
Epoch 28/70
200/200 [==============================] - ETA: 0s - loss: 0.0932 - iou_score:
0.8954 - f1-score: 0.9433
Epoch 28: val_loss improved from 0.20038 to 0.19969, saving model to
model_weights/20221118-232939_segmentation_best_weights.hdf5
200/200 [==============================] - 76s 378ms/step - loss: 0.0932 -
iou_score: 0.8954 - f1-score: 0.9433 - val_loss: 0.1997 - val_iou_score: 0.8100
- val_f1-score: 0.8813 - lr: 1.0000e-04
Epoch 29/70
200/200 [==============================] - ETA: 0s - loss: 0.0845 - iou_score:
0.9040 - f1-score: 0.9485
Epoch 29: val_loss did not improve from 0.19969
200/200 [==============================] - 74s 369ms/step - loss: 0.0845 -
iou_score: 0.9040 - f1-score: 0.9485 - val_loss: 0.3004 - val_iou_score: 0.6913
- val_f1-score: 0.7839 - lr: 1.0000e-04
Epoch 30/70
200/200 [==============================] - ETA: 0s - loss: 0.1041 - iou_score:
0.8888 - f1-score: 0.9389
Epoch 30: val_loss did not improve from 0.19969
200/200 [==============================] - 74s 370ms/step - loss: 0.1041 -
iou_score: 0.8888 - f1-score: 0.9389 - val_loss: 0.2410 - val_iou_score: 0.7565
- val_f1-score: 0.8404 - lr: 1.0000e-04
Epoch 31/70
200/200 [==============================] - ETA: 0s - loss: 0.1003 - iou_score:
0.8897 - f1-score: 0.9387
```

```
Epoch 31: val_loss did not improve from 0.19969
200/200 [==============================] - 74s 368ms/step - loss: 0.1003 -
iou_score: 0.8897 - f1-score: 0.9387 - val_loss: 0.2039 - val_iou_score: 0.8022
- val_f1-score: 0.8752 - lr: 1.0000e-04
Epoch 32/70
200/200 [==============================] - ETA: 0s - loss: 0.0992 - iou_score:
0.8906 - f1-score: 0.9404
Epoch 32: val_loss did not improve from 0.19969
200/200 [==============================] - 74s 369ms/step - loss: 0.0992 -
iou_score: 0.8906 - f1-score: 0.9404 - val_loss: 0.2639 - val_iou_score: 0.7394
- val_f1-score: 0.8293 - lr: 1.0000e-04
Epoch 33/70
200/200 [==============================] - ETA: 0s - loss: 0.0808 - iou_score:
0.9077 - f1-score: 0.9504
Epoch 33: val_loss improved from 0.19969 to 0.19224, saving model to
model_weights/20221118-232939_segmentation_best_weights.hdf5
200/200 [==============================] - 76s 380ms/step - loss: 0.0808 -
iou_score: 0.9077 - f1-score: 0.9504 - val_loss: 0.1922 - val_iou_score: 0.8124
- val_f1-score: 0.8833 - lr: 1.0000e-04
Epoch 34/70
200/200 [==============================] - ETA: 0s - loss: 0.0823 - iou_score:
0.9060 - f1-score: 0.9494
Epoch 34: val_loss did not improve from 0.19224
200/200 [==============================] - 74s 369ms/step - loss: 0.0823 -
iou_score: 0.9060 - f1-score: 0.9494 - val_loss: 0.2499 - val_iou_score: 0.7959
- val_f1-score: 0.8703 - lr: 1.0000e-04
Epoch 35/70
200/200 [==============================] - ETA: 0s - loss: 0.0748 - iou_score:
0.9135 - f1-score: 0.9538
Epoch 35: val_loss did not improve from 0.19224
200/200 [==============================] - 74s 370ms/step - loss: 0.0748 -
iou_score: 0.9135 - f1-score: 0.9538 - val_loss: 0.2034 - val_iou_score: 0.8003
- val_f1-score: 0.8759 - lr: 1.0000e-04
Epoch 36/70
200/200 [==============================] - ETA: 0s - loss: 0.0781 - iou_score:
0.9112 - f1-score: 0.9524
Epoch 36: val_loss did not improve from 0.19224
200/200 [==============================] - 74s 369ms/step - loss: 0.0781 -
iou_score: 0.9112 - f1-score: 0.9524 - val_loss: 0.2040 - val_iou_score: 0.8049
- val_f1-score: 0.8772 - lr: 1.0000e-04
Epoch 37/70
200/200 [==============================] - ETA: 0s - loss: 0.0790 - iou_score:
0.9095 - f1-score: 0.9517
Epoch 37: val_loss did not improve from 0.19224
200/200 [==============================] - 74s 369ms/step - loss: 0.0790 -
iou_score: 0.9095 - f1-score: 0.9517 - val_loss: 0.2276 - val_iou_score: 0.8069
- val_f1-score: 0.8806 - lr: 1.0000e-04
Epoch 38/70
```

```
200/200 [==============================] - ETA: 0s - loss: 0.0869 - iou_score:
0.9023 - f1-score: 0.9472
Epoch 38: val_loss did not improve from 0.19224
200/200 [==============================] - 74s 370ms/step - loss: 0.0869 -
iou_score: 0.9023 - f1-score: 0.9472 - val_loss: 0.2041 - val_iou_score: 0.8084
- val_f1-score: 0.8798 - lr: 1.0000e-04
Epoch 39/70
200/200 [==============================] - ETA: 0s - loss: 0.0861 - iou_score:
0.9019 - f1-score: 0.9473
Epoch 39: val_loss did not improve from 0.19224
200/200 [==============================] - 74s 369ms/step - loss: 0.0861 -
iou_score: 0.9019 - f1-score: 0.9473 - val_loss: 0.2070 - val_iou_score: 0.8033
- val_f1-score: 0.8748 - lr: 1.0000e-04
Epoch 40/70
200/200 [==============================] - ETA: 0s - loss: 0.0861 - iou_score:
0.9040 - f1-score: 0.9482
Epoch 40: val_loss did not improve from 0.19224
200/200 [==============================] - 74s 369ms/step - loss: 0.0861 -
iou_score: 0.9040 - f1-score: 0.9482 - val_loss: 0.2540 - val_iou_score: 0.7782
- val_f1-score: 0.8532 - lr: 1.0000e-04
Epoch 41/70
200/200 [==============================] - ETA: 0s - loss: 0.0875 - iou_score:
0.9003 - f1-score: 0.9461
Epoch 41: val_loss did not improve from 0.19224
200/200 [==============================] - 74s 370ms/step - loss: 0.0875 -
iou_score: 0.9003 - f1-score: 0.9461 - val_loss: 0.2503 - val_iou_score: 0.8000
- val_f1-score: 0.8753 - lr: 1.0000e-04
Epoch 42/70
200/200 [==============================] - ETA: 0s - loss: 0.0770 - iou_score:
0.9119 - f1-score: 0.9530
Epoch 42: val_loss did not improve from 0.19224
200/200 [==============================] - 74s 369ms/step - loss: 0.0770 -
iou_score: 0.9119 - f1-score: 0.9530 - val_loss: 0.2034 - val_iou_score: 0.8085
- val_f1-score: 0.8768 - lr: 1.0000e-04
Epoch 43/70
200/200 [==============================] - ETA: 0s - loss: 0.0705 - iou_score:
0.9192 - f1-score: 0.9571
Epoch 43: val_loss did not improve from 0.19224
200/200 [==============================] - 74s 368ms/step - loss: 0.0705 -
iou_score: 0.9192 - f1-score: 0.9571 - val_loss: 0.2026 - val_iou_score: 0.8116
- val_f1-score: 0.8799 - lr: 1.0000e-04
Epoch 44/70
200/200 [==============================] - ETA: 0s - loss: 0.0686 - iou_score:
0.9206 - f1-score: 0.9579
Epoch 44: val_loss did not improve from 0.19224
200/200 [==============================] - 74s 369ms/step - loss: 0.0686 -
iou_score: 0.9206 - f1-score: 0.9579 - val_loss: 0.2025 - val_iou_score: 0.8157
- val_f1-score: 0.8839 - lr: 1.0000e-05
```

```
Epoch 45/70
200/200 [==============================] - ETA: 0s - loss: 0.0648 - iou_score:
0.9240 - f1-score: 0.9595
Epoch 45: val_loss did not improve from 0.19224
200/200 [==============================] - 74s 368ms/step - loss: 0.0648 -
iou_score: 0.9240 - f1-score: 0.9595 - val_loss: 0.1942 - val_iou_score: 0.8185
- val_f1-score: 0.8852 - lr: 1.0000e-05
Epoch 46/70
200/200 [==============================] - ETA: 0s - loss: 0.0617 - iou_score:
0.9260 - f1-score: 0.9610
Epoch 46: val_loss did not improve from 0.19224
200/200 [==============================] - 74s 369ms/step - loss: 0.0617 -
iou_score: 0.9260 - f1-score: 0.9610 - val_loss: 0.1951 - val_iou_score: 0.8197
- val_f1-score: 0.8866 - lr: 1.0000e-05
Epoch 47/70
200/200 [==============================] - ETA: 0s - loss: 0.0606 - iou_score:
0.9286 - f1-score: 0.9624
Epoch 47: val_loss did not improve from 0.19224
200/200 [==============================] - 74s 369ms/step - loss: 0.0606 -
iou_score: 0.9286 - f1-score: 0.9624 - val_loss: 0.1941 - val_iou_score: 0.8210
- val_f1-score: 0.8877 - lr: 1.0000e-05
Epoch 48/70
200/200 [==============================] - ETA: 0s - loss: 0.0604 - iou_score:
0.9275 - f1-score: 0.9618
Epoch 48: val_loss did not improve from 0.19224
200/200 [==============================] - 74s 368ms/step - loss: 0.0604 -
iou_score: 0.9275 - f1-score: 0.9618 - val_loss: 0.1926 - val_iou_score: 0.8222
- val_f1-score: 0.8888 - lr: 1.0000e-05
Epoch 49/70
200/200 [==============================] - ETA: 0s - loss: 0.0595 - iou_score:
0.9291 - f1-score: 0.9627
Epoch 49: val_loss improved from 0.19224 to 0.19052, saving model to
model_weights/20221118-232939_segmentation_best_weights.hdf5
200/200 [==============================] - 75s 375ms/step - loss: 0.0595 -
iou_score: 0.9291 - f1-score: 0.9627 - val_loss: 0.1905 - val_iou_score: 0.8229
- val_f1-score: 0.8893 - lr: 1.0000e-05
Epoch 50/70
200/200 [==============================] - ETA: 0s - loss: 0.0602 - iou_score:
0.9285 - f1-score: 0.9624
Epoch 50: val_loss did not improve from 0.19052
200/200 [==============================] - 74s 369ms/step - loss: 0.0602 -
iou_score: 0.9285 - f1-score: 0.9624 - val_loss: 0.1922 - val_iou_score: 0.8215
- val_f1-score: 0.8880 - lr: 1.0000e-05
Epoch 51/70
200/200 [==============================] - ETA: 0s - loss: 0.0559 - iou_score:
0.9332 - f1-score: 0.9650
Epoch 51: val_loss improved from 0.19052 to 0.19018, saving model to
model_weights/20221118-232939_segmentation_best_weights.hdf5
```

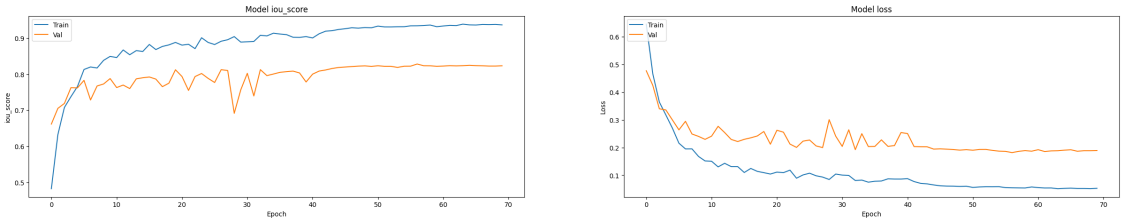
```
200/200 [==============================] - 75s 374ms/step - loss: 0.0559 -
iou_score: 0.9332 - f1-score: 0.9650 - val_loss: 0.1902 - val_iou_score: 0.8232
- val_f1-score: 0.8890 - lr: 1.0000e-05
Epoch 52/70
200/200 [==============================] - ETA: 0s - loss: 0.0576 - iou_score:
0.9309 - f1-score: 0.9638
Epoch 52: val_loss did not improve from 0.19018
200/200 [==============================] - 74s 370ms/step - loss: 0.0576 -
iou_score: 0.9309 - f1-score: 0.9638 - val_loss: 0.1929 - val_iou_score: 0.8217
- val_f1-score: 0.8877 - lr: 1.0000e-05
Epoch 53/70
200/200 [==============================] - ETA: 0s - loss: 0.0584 - iou_score:
0.9309 - f1-score: 0.9636
Epoch 53: val_loss did not improve from 0.19018
200/200 [==============================] - 78s 391ms/step - loss: 0.0584 -
iou_score: 0.9309 - f1-score: 0.9636 - val_loss: 0.1930 - val_iou_score: 0.8216
- val_f1-score: 0.8879 - lr: 1.0000e-05
Epoch 54/70
200/200 [==============================] - ETA: 0s - loss: 0.0583 - iou_score:
0.9314 - f1-score: 0.9639
Epoch 54: val_loss improved from 0.19018 to 0.18956, saving model to
model_weights/20221118-232939_segmentation_best_weights.hdf5
200/200 [==============================] - 78s 389ms/step - loss: 0.0583 -
iou_score: 0.9314 - f1-score: 0.9639 - val_loss: 0.1896 - val_iou_score: 0.8186
- val_f1-score: 0.8850 - lr: 1.0000e-05
Epoch 55/70
200/200 [==============================] - ETA: 0s - loss: 0.0585 - iou_score:
0.9315 - f1-score: 0.9639
Epoch 55: val_loss improved from 0.18956 to 0.18682, saving model to
model_weights/20221118-232939_segmentation_best_weights.hdf5
200/200 [==============================] - 75s 375ms/step - loss: 0.0585 -
iou_score: 0.9315 - f1-score: 0.9639 - val_loss: 0.1868 - val_iou_score: 0.8219
- val_f1-score: 0.8875 - lr: 1.0000e-05
Epoch 56/70
200/200 [==============================] - ETA: 0s - loss: 0.0551 - iou_score:
0.9339 - f1-score: 0.9654
Epoch 56: val_loss improved from 0.18682 to 0.18605, saving model to
model_weights/20221118-232939_segmentation_best_weights.hdf5
200/200 [==============================] - 74s 370ms/step - loss: 0.0551 -
iou_score: 0.9339 - f1-score: 0.9654 - val_loss: 0.1861 - val_iou_score: 0.8221
- val_f1-score: 0.8881 - lr: 1.0000e-05
Epoch 57/70
200/200 [==============================] - ETA: 0s - loss: 0.0546 - iou_score:
0.9341 - f1-score: 0.9654
Epoch 57: val_loss improved from 0.18605 to 0.18152, saving model to
model_weights/20221118-232939_segmentation_best_weights.hdf5
200/200 [==============================] - 78s 387ms/step - loss: 0.0546 -
iou_score: 0.9341 - f1-score: 0.9654 - val_loss: 0.1815 - val_iou_score: 0.8280
```

```
- val_f1-score: 0.8931 - lr: 1.0000e-05
Epoch 58/70
200/200 [==============================] - ETA: 0s - loss: 0.0542 - iou_score:
0.9349 - f1-score: 0.9659
Epoch 58: val_loss did not improve from 0.18152
200/200 [==============================] - 71s 355ms/step - loss: 0.0542 -
iou_score: 0.9349 - f1-score: 0.9659 - val_loss: 0.1861 - val_iou_score: 0.8233
- val_f1-score: 0.8897 - lr: 1.0000e-05
Epoch 59/70
200/200 [==============================] - ETA: 0s - loss: 0.0539 - iou_score:
0.9361 - f1-score: 0.9666
Epoch 59: val_loss did not improve from 0.18152
200/200 [==============================] - 74s 369ms/step - loss: 0.0539 -
iou_score: 0.9361 - f1-score: 0.9666 - val_loss: 0.1888 - val_iou_score: 0.8231
- val_f1-score: 0.8900 - lr: 1.0000e-05
Epoch 60/70
200/200 [==============================] - ETA: 0s - loss: 0.0575 - iou_score:
0.9315 - f1-score: 0.9640
Epoch 60: val_loss did not improve from 0.18152
200/200 [==============================] - 74s 369ms/step - loss: 0.0575 -
iou_score: 0.9315 - f1-score: 0.9640 - val_loss: 0.1866 - val_iou_score: 0.8217
- val_f1-score: 0.8883 - lr: 1.0000e-05
Epoch 61/70
200/200 [==============================] - ETA: 0s - loss: 0.0552 - iou_score:
0.9335 - f1-score: 0.9651
Epoch 61: val_loss did not improve from 0.18152
200/200 [==============================] - 74s 370ms/step - loss: 0.0552 -
iou_score: 0.9335 - f1-score: 0.9651 - val_loss: 0.1921 - val_iou_score: 0.8223
- val_f1-score: 0.8888 - lr: 1.0000e-05
Epoch 62/70
200/200 [==============================] - ETA: 0s - loss: 0.0539 - iou_score:
0.9352 - f1-score: 0.9661
Epoch 62: val_loss did not improve from 0.18152
200/200 [==============================] - 74s 370ms/step - loss: 0.0539 -
iou_score: 0.9352 - f1-score: 0.9661 - val_loss: 0.1854 - val_iou_score: 0.8234
- val_f1-score: 0.8894 - lr: 1.0000e-05
Epoch 63/70
200/200 [==============================] - ETA: 0s - loss: 0.0539 - iou_score:
0.9346 - f1-score: 0.9658
Epoch 63: val_loss did not improve from 0.18152
200/200 [==============================] - 74s 369ms/step - loss: 0.0539 -
iou_score: 0.9346 - f1-score: 0.9658 - val_loss: 0.1878 - val_iou_score: 0.8228
- val_f1-score: 0.8890 - lr: 1.0000e-05
Epoch 64/70
200/200 [==============================] - ETA: 0s - loss: 0.0512 - iou_score:
0.9386 - f1-score: 0.9680
Epoch 64: val_loss did not improve from 0.18152
200/200 [==============================] - 74s 369ms/step - loss: 0.0512 -
```

```
iou_score: 0.9386 - f1-score: 0.9680 - val_loss: 0.1884 - val_iou_score: 0.8234
- val_f1-score: 0.8896 - lr: 1.0000e-05
Epoch 65/70
200/200 [==============================] - ETA: 0s - loss: 0.0524 - iou_score:
0.9366 - f1-score: 0.9669
Epoch 65: val_loss did not improve from 0.18152
200/200 [==============================] - 74s 370ms/step - loss: 0.0524 -
iou_score: 0.9366 - f1-score: 0.9669 - val_loss: 0.1904 - val_iou_score: 0.8244
- val_f1-score: 0.8913 - lr: 1.0000e-05
Epoch 66/70
200/200 [==============================] - ETA: 0s - loss: 0.0533 - iou_score:
0.9362 - f1-score: 0.9665
Epoch 66: val_loss did not improve from 0.18152
200/200 [==============================] - 74s 369ms/step - loss: 0.0533 -
iou_score: 0.9362 - f1-score: 0.9665 - val_loss: 0.1919 - val_iou_score: 0.8235
- val_f1-score: 0.8905 - lr: 1.0000e-05
Epoch 67/70
200/200 [==============================] - ETA: 0s - loss: 0.0519 - iou_score:
0.9377 - f1-score: 0.9675
Epoch 67: val_loss did not improve from 0.18152
200/200 [==============================] - 74s 369ms/step - loss: 0.0519 -
iou_score: 0.9377 - f1-score: 0.9675 - val_loss: 0.1864 - val_iou_score: 0.8231
- val_f1-score: 0.8893 - lr: 1.0000e-05
Epoch 68/70
200/200 [==============================] - ETA: 0s - loss: 0.0519 - iou_score:
0.9374 - f1-score: 0.9673
Epoch 68: val_loss did not improve from 0.18152
200/200 [==============================] - 74s 370ms/step - loss: 0.0519 -
iou_score: 0.9374 - f1-score: 0.9673 - val_loss: 0.1884 - val_iou_score: 0.8223
- val_f1-score: 0.8882 - lr: 1.0000e-06
Epoch 69/70
200/200 [==============================] - ETA: 0s - loss: 0.0514 - iou_score:
0.9378 - f1-score: 0.9675
Epoch 69: val_loss did not improve from 0.18152
200/200 [==============================] - 74s 369ms/step - loss: 0.0514 -
iou_score: 0.9378 - f1-score: 0.9675 - val_loss: 0.1884 - val_iou_score: 0.8223
- val_f1-score: 0.8883 - lr: 1.0000e-06
Epoch 70/70
200/200 [==============================] - ETA: 0s - loss: 0.0526 - iou_score:
0.9366 - f1-score: 0.9668
Epoch 70: val_loss did not improve from 0.18152
200/200 [==============================] - 74s 369ms/step - loss: 0.0526 -
iou_score: 0.9366 - f1-score: 0.9668 - val_loss: 0.1890 - val_iou_score: 0.8231
- val_f1-score: 0.8886 - lr: 1.0000e-06
```

[ ]:
```
# Plot training & validation iou_score values
plt.figure(figsize=(30, 5))
```

```
plt.subplot(121)
plt.plot(history.history['iou_score'])
plt.plot(history.history['val_iou_score'])
plt.title('Model iou_score')
plt.ylabel('iou_score')
plt.xlabel('Epoch')
plt.legend(['Train', 'Val'], loc='upper left')

# Plot training & validation loss values
plt.subplot(122)
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Val'], loc='upper left')
plt.show()
```



4 Model Evaluation

[ ]:
```
test_dataset = Dataset(
    x_test_dir,
    y_test_dir,
    classes=CLASSES,
    #augmentation=get_validation_augmentation(),
    preprocessing=get_preprocessing(preprocess_input),
)

test_dataloader = Dataloder(test_dataset, batch_size=1, shuffle=False)
```

[ ]:
```
print(date_actual)
```

```
20221118-232939
```

[ ]:
```
# load best weights
model.load_weights(f'model_weights/{date_actual}_segmentation_best_weights.
 ↪hdf5')
```

```
#model.load_weights("best_model_weights/seg_model_resnet_50_1.hdf5")
```

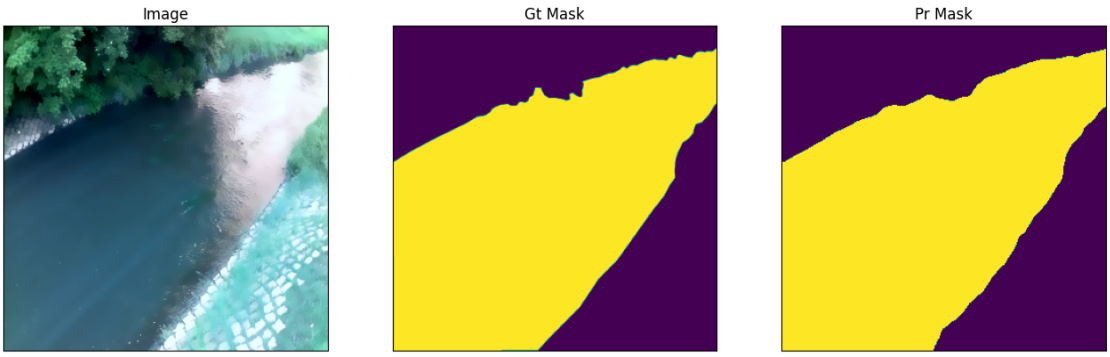
[ ]:
```
scores = model.evaluate(test_dataloader)

print("Loss: {:.5}".format(scores[0]))
for metric, value in zip(metrics, scores[1:]):
    print("mean {}: {:.5}".format(metric.__name__, value))
```

```
111/111 [==============================] - 6s 42ms/step - loss: -74.0869 -
iou_score: 0.9106 - f1-score: 0.8272
Loss: -74.087
mean iou_score: 0.9106
mean f1-score: 0.82723
```

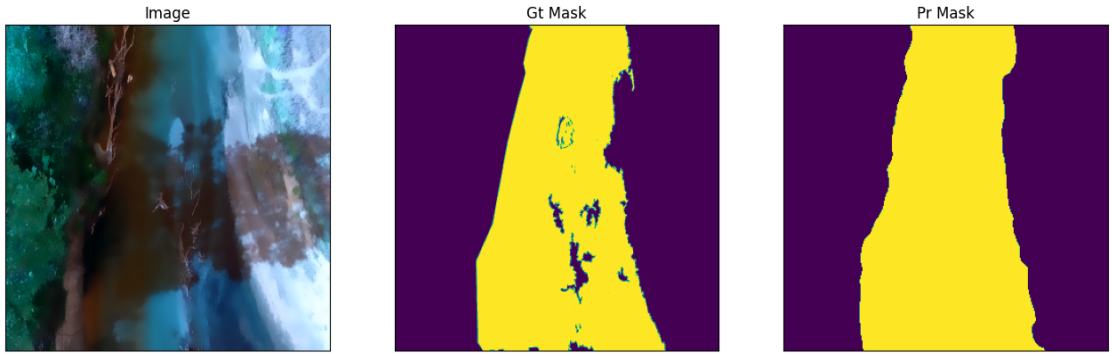
5 Visualization of results on test dataset

[ ]:
```
n = 5
ids = np.random.choice(np.arange(len(test_dataset)), size=n)

for i in ids:

    image, gt_mask = test_dataset[i]
    image = np.expand_dims(image, axis=0)
    pr_mask = model.predict(image).round()

    visualize(
        image=denormalize(image.squeeze()),
        gt_mask=gt_mask.squeeze(),
        pr_mask=pr_mask.squeeze(),
    )
```

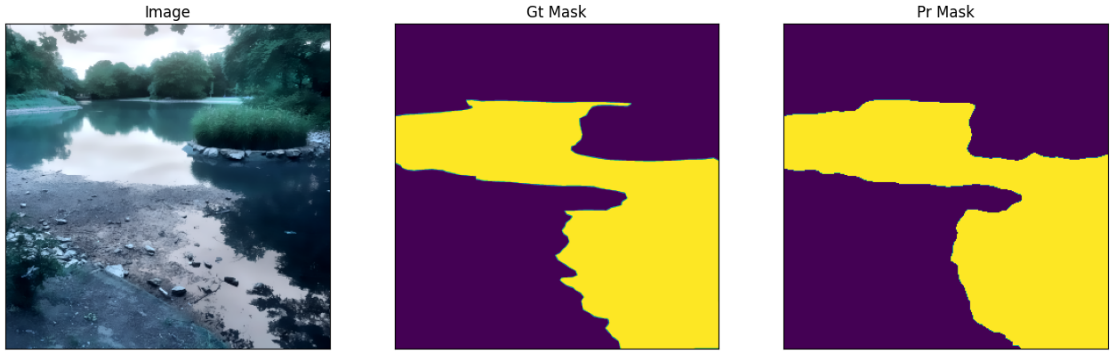
```
1/1 [==============================] - 0s 42ms/step
```



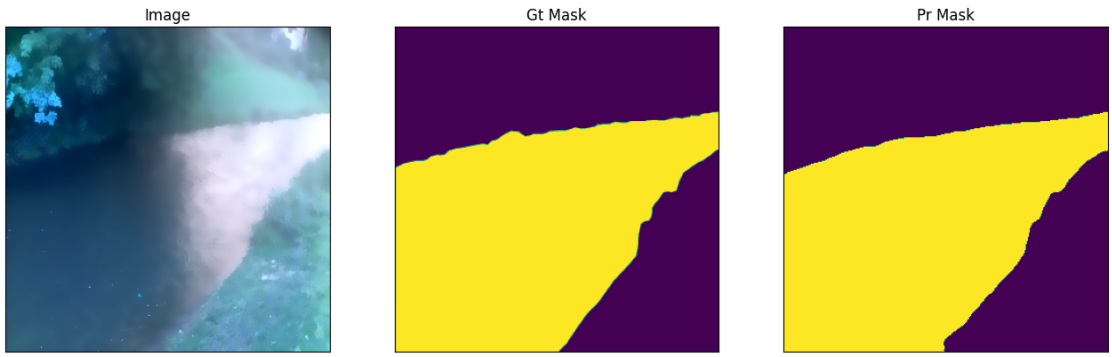
```
1/1 [==============================] - 0s 45ms/step
```



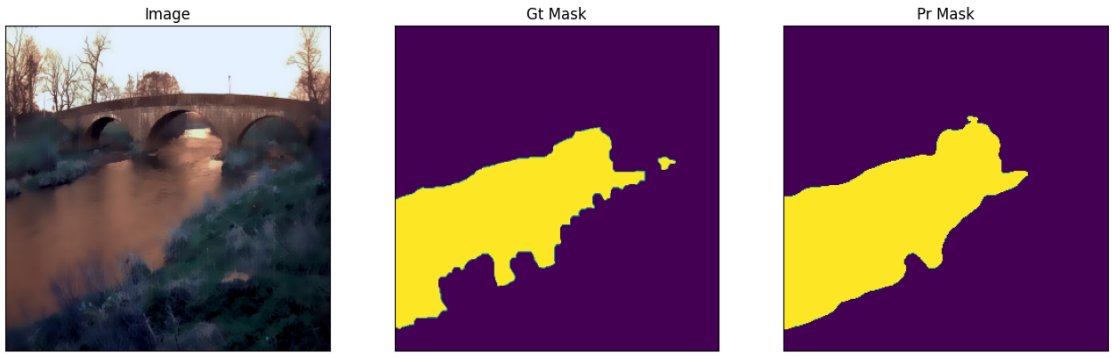

```
1/1 [==============================] - 0s 42ms/step
```




```
1/1 [==============================] - 0s 42ms/step
```




```
1/1 [==============================] - 0s 41ms/step
```



5.1 Visualize with test dataset of images for hydrology project

[ ]:
```
# define heavy augmentations
def get_image_fixing():
    train_transform = [
        A.OneOf(
            [
                A.RandomBrightness(limit=0.4, p=1),
            ],
            p=0.9,
        ),

        A.OneOf(
            [
                A.Sharpen(alpha=(0.2, 0.3), p=1),
            ],
            p=0.9,
        ),

        A.OneOf(
            [
                A.RandomContrast(p=1),
                #A.HueSaturationValue(p=1),
            ],
            p=0.9,
        ),
    ]
    return A.Compose(train_transform)
```

[ ]:
```
test_h_dataset = Dataset(
    "river_segmented_hydrology/images",
    "river_segmented_hydrology/masks",
    classes=CLASSES,
```

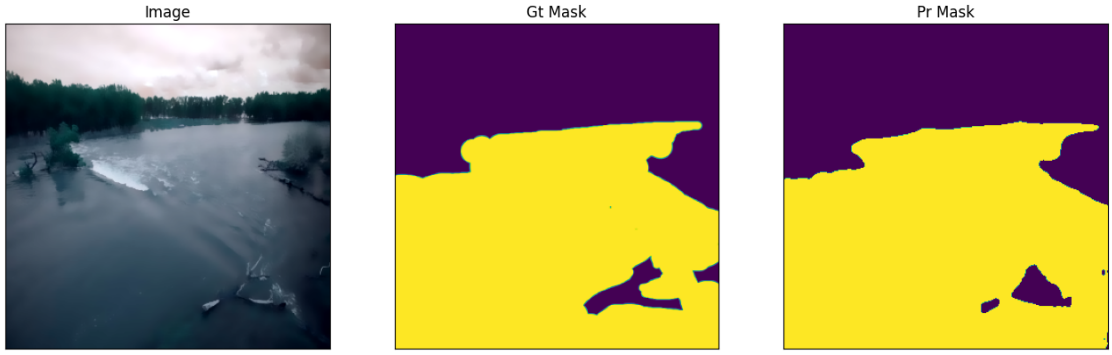
```
    #augmentation=get_image_fixing(),
    preprocessing=get_preprocessing(preprocess_input),
)

test_h_dataloader = Dataloder(test_h_dataset, batch_size=1, shuffle=False)
```

[ ]:
```
n = 10
#ids = np.random.choice(np.arange(len(test_h_dataset)), size=n)

for i in range(0, len(test_h_dataset)):

    image, gt_mask = test_h_dataset[i]
    image = np.expand_dims(image, axis=0)
    pr_mask = model.predict(image).round()

    visualize(
        image=denormalize(image.squeeze()),
        gt_mask=gt_mask.squeeze(),
        pr_mask=pr_mask.squeeze(),
    )
```

```
1/1 [==============================] - 0s 41ms/step
```

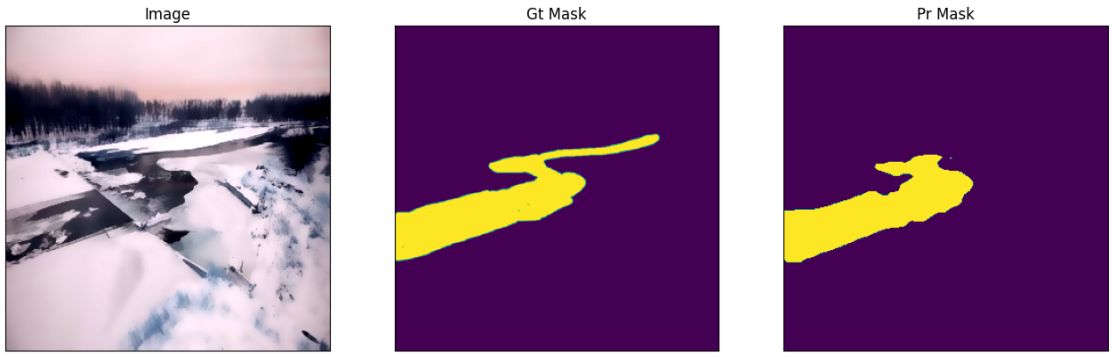

```
1/1 [==============================] - 0s 43ms/step
```


```
1/1 [==============================] - 0s 42ms/step
```

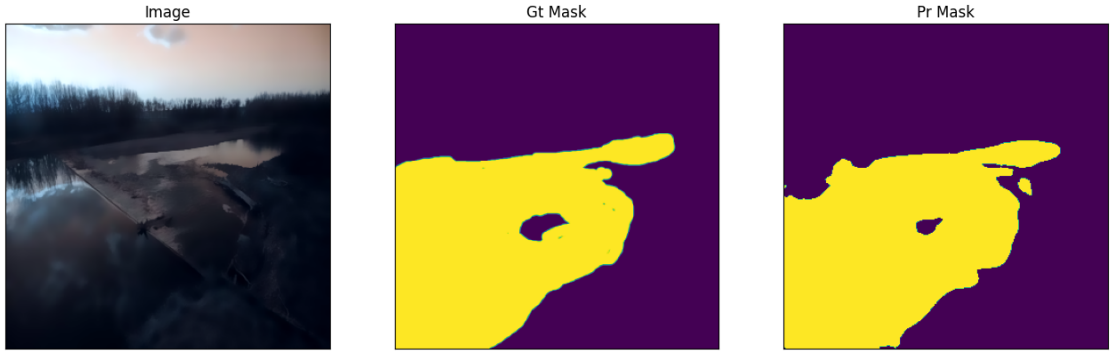



```
1/1 [==============================] - 0s 40ms/step
```

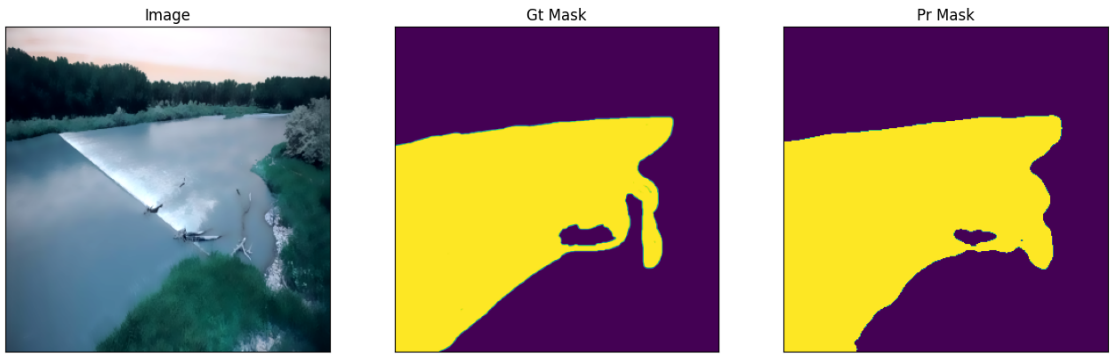

```
1/1 [==============================] - 0s 40ms/step
```



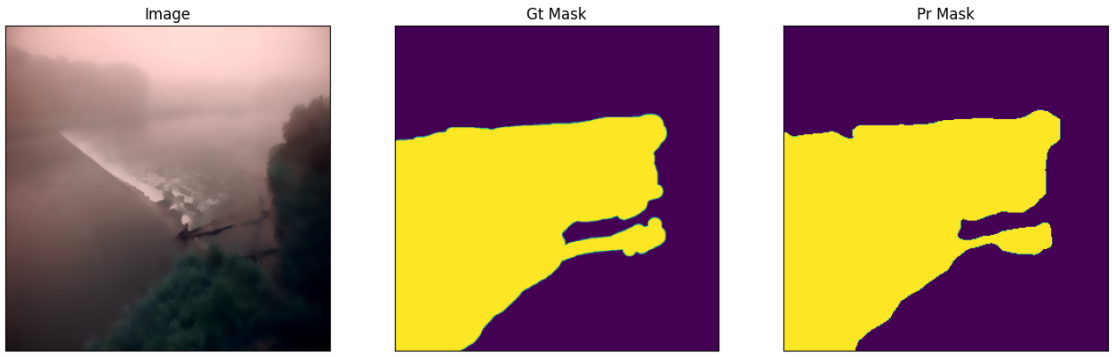

```
1/1 [==============================] - 0s 44ms/step
```




```
1/1 [==============================] - 0s 41ms/step
```

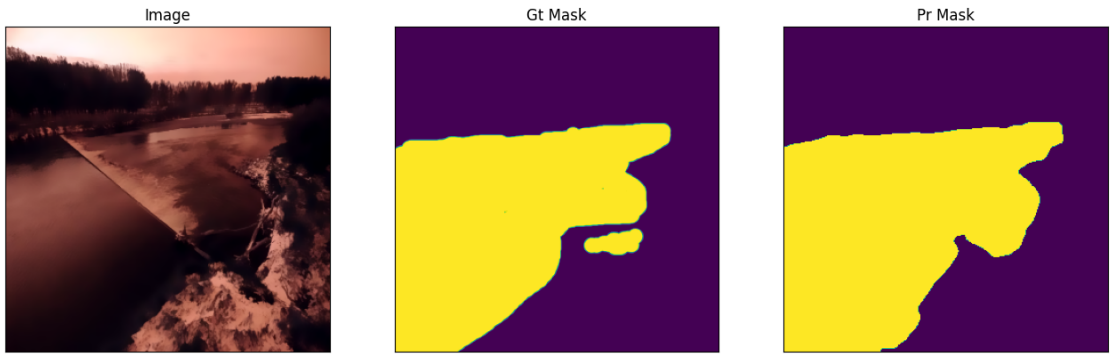



```
1/1 [==============================] - 0s 39ms/step
```




```
1/1 [==============================] - 0s 40ms/step
```


```
1/1 [==============================] - 0s 41ms/step
```




```
1/1 [==============================] - 0s 40ms/step
```

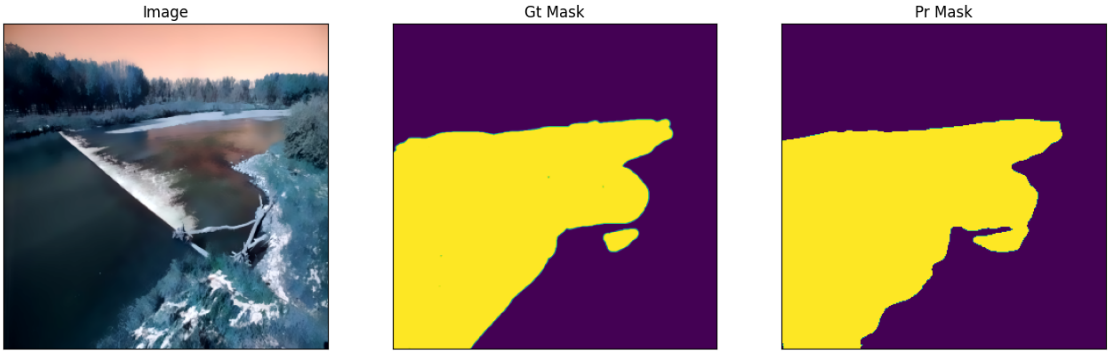


Image
Gt Mask
Pr Mask