binary segmentation

November 22, 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
```

```
2022-11-22 12:29:52.432895: 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.
2022-11-22 12:29:52.547288: 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
2022-11-22 12:29:53.011262: 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/
2022-11-22 12:29:53.011319: 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/
2022-11-22 12:29:53.011324: 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)
```

```
2022-11-22 12:29:53.886989: 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
2022-11-22 12:29:53.892455: 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
2022-11-22 12:29:53.892722: 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, 9, 35, 19)

    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']
```

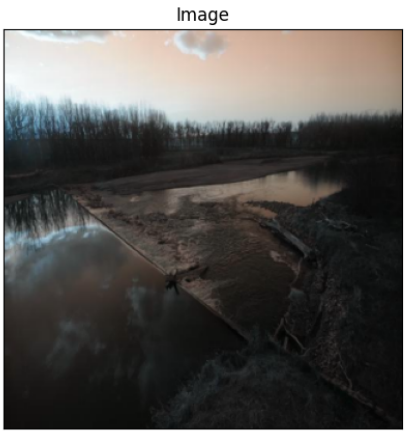
```
def __init__(
        self,
        images_dir,
        masks_dir,
        classes=None,
        augmentation=None,
        preprocessing=None,
):
    self.ids = os.listdir(images_dir)
    self.images_fps = [str(i) for i in glob(images_dir + '/*')]
    self.masks_fps = [str(i) for i in glob(masks_dir + '/*')]
    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], cv2.COLOR_BGR2RGB)
    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.

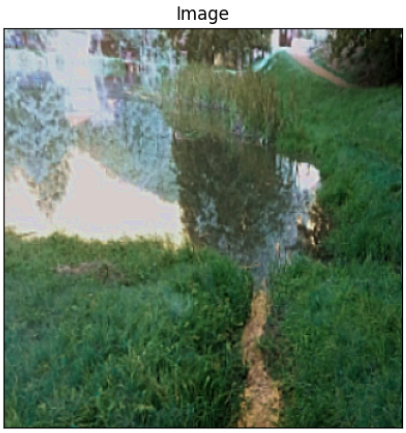
[ ]:
```
import albumentations as A
```

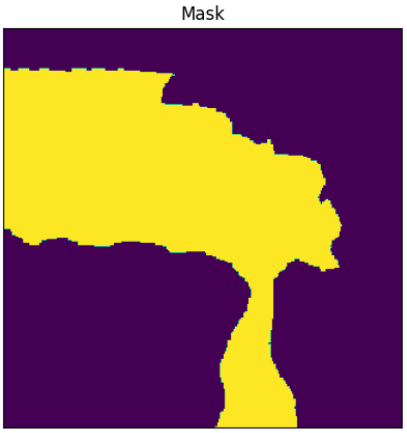
[ ]:
```
def round_clip_0_1(x, **kwargs):
    return x.round().clip(0, 1)

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

        A.HorizontalFlip(p=0.5),

        A.ShiftScaleRotate(scale_limit=0.5, rotate_limit=0.3, 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=0.3),
        A.Perspective(p=0.5),

        A.OneOf(
            [
                A.CLAHE(p=1),
                A.RandomBrightness(limit=0.4, p=1),
                A.RandomGamma(p=1),
            ],
            p=0.9,
        ),

        A.OneOf(
            [
                A.Sharpen(p=1),
                A.Blur(blur_limit=2, 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'
BACKBONE = 'seresnext101'
BATCH_SIZE = 4
CLASSES = [1]
LR = 0.0001
EPOCHS = 100

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)
```

```
2022-11-22 12:29:54.719025: 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.
2022-11-22 12:29:54.719659: 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
2022-11-22 12:29:54.719891: 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
2022-11-22 12:29:54.720093: 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
2022-11-22 12:29:55.119561: 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
2022-11-22 12:29:55.119786: 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
2022-11-22 12:29:55.119994: 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
2022-11-22 12:29:55.120160: I
tensorflow/core/common_runtime/gpu/gpu_device.cc:1616] Created device
/job:localhost/replica:0/task:0/device:GPU:0 with 4191 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=10)

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(patience=6),
]
```

[ ]:
```
# 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 1/100

2022-11-22 12:31:03.854319: I tensorflow/stream_executor/cuda/cuda_dnn.cc:384]
Loaded cuDNN version 8100
2022-11-22 12:31:05.059322: I
tensorflow/core/platform/default/subprocess.cc:304] Start cannot spawn child
process: No such file or directory
2022-11-22 12:31:05.060037: I
tensorflow/core/platform/default/subprocess.cc:304] Start cannot spawn child
process: No such file or directory
2022-11-22 12:31:05.060049: W tensorflow/stream_executor/gpu/asm_compiler.cc:80]
Couldn't get ptxas version string: INTERNAL: Couldn't invoke ptxas --version
2022-11-22 12:31:05.060840: I
tensorflow/core/platform/default/subprocess.cc:304] Start cannot spawn child
process: No such file or directory
2022-11-22 12:31:05.060876: 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.
2022-11-22 12:31:06.107728: 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.
2022-11-22 12:31:06.107758: 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.
2022-11-22 12:31:06.143780: 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.
2022-11-22 12:31:06.143803: 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.
2022-11-22 12:31:06.201498: 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.
2022-11-22 12:31:06.201524: 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.
2022-11-22 12:31:06.228718: 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.
2022-11-22 12:31:06.228738: 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.
2022-11-22 12:31:06.288131: 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.
2022-11-22 12:31:06.288153: 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: 0.5224 - iou_score:
0.5265 - f1-score: 0.6705
Epoch 1: val_loss improved from inf to 0.44684, saving model to
model_weights/20221122-123004_segmentation_best_weights.hdf5

2022-11-22 12:33:12.045490: W
```

```
tensorflow/core/framework/cpu_allocator_impl.cc:82] Allocation of 1698693120
exceeds 10% of free system memory.

200/200 [==============================] - 187s 614ms/step - loss: 0.5224 -
iou_score: 0.5265 - f1-score: 0.6705 - val_loss: 0.4468 - val_iou_score: 0.6970
- val_f1-score: 0.8000 - lr: 1.0000e-04
Epoch 2/100
200/200 [==============================] - ETA: 0s - loss: 0.3571 - iou_score:
0.6857 - f1-score: 0.8035
Epoch 2: val_loss improved from 0.44684 to 0.39547, saving model to
model_weights/20221122-123004_segmentation_best_weights.hdf5

2022-11-22 12:35:08.152708: W
tensorflow/core/framework/cpu_allocator_impl.cc:82] Allocation of 1698693120
exceeds 10% of free system memory.

200/200 [==============================] - 116s 580ms/step - loss: 0.3571 -
iou_score: 0.6857 - f1-score: 0.8035 - val_loss: 0.3955 - val_iou_score: 0.7125
- val_f1-score: 0.8104 - lr: 1.0000e-04
Epoch 3/100
200/200 [==============================] - ETA: 0s - loss: 0.2737 - iou_score:
0.7524 - f1-score: 0.8522
Epoch 3: val_loss improved from 0.39547 to 0.26032, saving model to
model_weights/20221122-123004_segmentation_best_weights.hdf5

2022-11-22 12:37:06.881657: W
tensorflow/core/framework/cpu_allocator_impl.cc:82] Allocation of 1698693120
exceeds 10% of free system memory.

200/200 [==============================] - 119s 593ms/step - loss: 0.2737 -
iou_score: 0.7524 - f1-score: 0.8522 - val_loss: 0.2603 - val_iou_score: 0.7591
- val_f1-score: 0.8463 - lr: 1.0000e-04
Epoch 4/100
200/200 [==============================] - ETA: 0s - loss: 0.2286 - iou_score:
0.7876 - f1-score: 0.8750
Epoch 4: val_loss did not improve from 0.26032

2022-11-22 12:39:01.140834: W
tensorflow/core/framework/cpu_allocator_impl.cc:82] Allocation of 1698693120
exceeds 10% of free system memory.

200/200 [==============================] - 114s 571ms/step - loss: 0.2286 -
iou_score: 0.7876 - f1-score: 0.8750 - val_loss: 0.2631 - val_iou_score: 0.7808
- val_f1-score: 0.8574 - lr: 1.0000e-04
Epoch 5/100
200/200 [==============================] - ETA: 0s - loss: 0.1955 - iou_score:
0.8108 - f1-score: 0.8901
Epoch 5: val_loss improved from 0.26032 to 0.22268, saving model to
model_weights/20221122-123004_segmentation_best_weights.hdf5

2022-11-22 12:40:57.607961: W
tensorflow/core/framework/cpu_allocator_impl.cc:82] Allocation of 1698693120
```

```
exceeds 10% of free system memory.
200/200 [==============================] - 116s 582ms/step - loss: 0.1955 -
iou_score: 0.8108 - f1-score: 0.8901 - val_loss: 0.2227 - val_iou_score: 0.8042
- val_f1-score: 0.8770 - lr: 1.0000e-04
Epoch 6/100
200/200 [==============================] - ETA: 0s - loss: 0.1841 - iou_score:
0.8196 - f1-score: 0.8959
Epoch 6: val_loss improved from 0.22268 to 0.20462, saving model to
model_weights/20221122-123004_segmentation_best_weights.hdf5
200/200 [==============================] - 115s 577ms/step - loss: 0.1841 -
iou_score: 0.8196 - f1-score: 0.8959 - val_loss: 0.2046 - val_iou_score: 0.7962
- val_f1-score: 0.8717 - lr: 1.0000e-04
Epoch 7/100
200/200 [==============================] - ETA: 0s - loss: 0.1787 - iou_score:
0.8243 - f1-score: 0.8991
Epoch 7: val_loss did not improve from 0.20462
200/200 [==============================] - 114s 568ms/step - loss: 0.1787 -
iou_score: 0.8243 - f1-score: 0.8991 - val_loss: 0.2237 - val_iou_score: 0.7926
- val_f1-score: 0.8677 - lr: 1.0000e-04
Epoch 8/100
200/200 [==============================] - ETA: 0s - loss: 0.1509 - iou_score:
0.8472 - f1-score: 0.9133
Epoch 8: val_loss improved from 0.20462 to 0.19597, saving model to
model_weights/20221122-123004_segmentation_best_weights.hdf5
200/200 [==============================] - 116s 579ms/step - loss: 0.1509 -
iou_score: 0.8472 - f1-score: 0.9133 - val_loss: 0.1960 - val_iou_score: 0.8091
- val_f1-score: 0.8816 - lr: 1.0000e-04
Epoch 9/100
200/200 [==============================] - ETA: 0s - loss: 0.1563 - iou_score:
0.8411 - f1-score: 0.9099
Epoch 9: val_loss did not improve from 0.19597
200/200 [==============================] - 113s 566ms/step - loss: 0.1563 -
iou_score: 0.8411 - f1-score: 0.9099 - val_loss: 0.2070 - val_iou_score: 0.8091
- val_f1-score: 0.8817 - lr: 1.0000e-04
Epoch 10/100
200/200 [==============================] - ETA: 0s - loss: 0.1448 - iou_score:
0.8523 - f1-score: 0.9171
Epoch 10: val_loss improved from 0.19597 to 0.18017, saving model to
model_weights/20221122-123004_segmentation_best_weights.hdf5
200/200 [==============================] - 116s 579ms/step - loss: 0.1448 -
iou_score: 0.8523 - f1-score: 0.9171 - val_loss: 0.1802 - val_iou_score: 0.8236
- val_f1-score: 0.8917 - lr: 1.0000e-04
Epoch 11/100
200/200 [==============================] - ETA: 0s - loss: 0.1468 - iou_score:
0.8468 - f1-score: 0.9129
Epoch 11: val_loss did not improve from 0.18017
200/200 [==============================] - 113s 567ms/step - loss: 0.1468 -
```

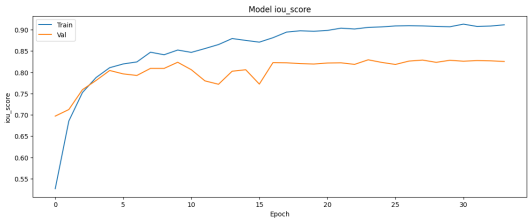
```
iou_score: 0.8468 - f1-score: 0.9129 - val_loss: 0.2029 - val_iou_score: 0.8060
- val_f1-score: 0.8779 - lr: 1.0000e-04
Epoch 12/100
200/200 [==============================] - ETA: 0s - loss: 0.1399 - iou_score:
0.8557 - f1-score: 0.9184
Epoch 12: val_loss did not improve from 0.18017
200/200 [==============================] - 113s 566ms/step - loss: 0.1399 -
iou_score: 0.8557 - f1-score: 0.9184 - val_loss: 0.2220 - val_iou_score: 0.7799
- val_f1-score: 0.8581 - lr: 1.0000e-04
Epoch 13/100
200/200 [==============================] - ETA: 0s - loss: 0.1265 - iou_score:
0.8652 - f1-score: 0.9250
Epoch 13: val_loss did not improve from 0.18017
200/200 [==============================] - 113s 567ms/step - loss: 0.1265 -
iou_score: 0.8652 - f1-score: 0.9250 - val_loss: 0.2571 - val_iou_score: 0.7719
- val_f1-score: 0.8526 - lr: 1.0000e-04
Epoch 14/100
200/200 [==============================] - ETA: 0s - loss: 0.1127 - iou_score:
0.8792 - f1-score: 0.9336
Epoch 14: val_loss did not improve from 0.18017
200/200 [==============================] - 113s 566ms/step - loss: 0.1127 -
iou_score: 0.8792 - f1-score: 0.9336 - val_loss: 0.2014 - val_iou_score: 0.8025
- val_f1-score: 0.8756 - lr: 1.0000e-04
Epoch 15/100
200/200 [==============================] - ETA: 0s - loss: 0.1163 - iou_score:
0.8749 - f1-score: 0.9311
Epoch 15: val_loss did not improve from 0.18017
200/200 [==============================] - 113s 566ms/step - loss: 0.1163 -
iou_score: 0.8749 - f1-score: 0.9311 - val_loss: 0.1967 - val_iou_score: 0.8059
- val_f1-score: 0.8791 - lr: 1.0000e-04
Epoch 16/100
200/200 [==============================] - ETA: 0s - loss: 0.1183 - iou_score:
0.8708 - f1-score: 0.9286
Epoch 16: val_loss did not improve from 0.18017
200/200 [==============================] - 113s 567ms/step - loss: 0.1183 -
iou_score: 0.8708 - f1-score: 0.9286 - val_loss: 0.2222 - val_iou_score: 0.7722
- val_f1-score: 0.8495 - lr: 1.0000e-04
Epoch 17/100
200/200 [==============================] - ETA: 0s - loss: 0.1092 - iou_score:
0.8812 - f1-score: 0.9345
Epoch 17: val_loss improved from 0.18017 to 0.18014, saving model to
model_weights/20221122-123004_segmentation_best_weights.hdf5
200/200 [==============================] - 115s 574ms/step - loss: 0.1092 -
iou_score: 0.8812 - f1-score: 0.9345 - val_loss: 0.1801 - val_iou_score: 0.8226
- val_f1-score: 0.8900 - lr: 1.0000e-05
Epoch 18/100
200/200 [==============================] - ETA: 0s - loss: 0.0963 - iou_score:
0.8944 - f1-score: 0.9427
```

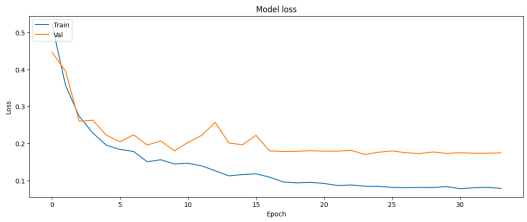
```
Epoch 18: val_loss improved from 0.18014 to 0.17867, saving model to
model_weights/20221122-123004_segmentation_best_weights.hdf5
200/200 [==============================] - 117s 584ms/step - loss: 0.0963 -
iou_score: 0.8944 - f1-score: 0.9427 - val_loss: 0.1787 - val_iou_score: 0.8223
- val_f1-score: 0.8892 - lr: 1.0000e-05
Epoch 19/100
200/200 [==============================] - ETA: 0s - loss: 0.0938 - iou_score:
0.8975 - f1-score: 0.9441
Epoch 19: val_loss did not improve from 0.17867
200/200 [==============================] - 113s 565ms/step - loss: 0.0938 -
iou_score: 0.8975 - f1-score: 0.9441 - val_loss: 0.1791 - val_iou_score: 0.8204
- val_f1-score: 0.8871 - lr: 1.0000e-05
Epoch 20/100
200/200 [==============================] - ETA: 0s - loss: 0.0952 - iou_score:
0.8963 - f1-score: 0.9436
Epoch 20: val_loss did not improve from 0.17867
200/200 [==============================] - 113s 566ms/step - loss: 0.0952 -
iou_score: 0.8963 - f1-score: 0.9436 - val_loss: 0.1806 - val_iou_score: 0.8194
- val_f1-score: 0.8865 - lr: 1.0000e-05
Epoch 21/100
200/200 [==============================] - ETA: 0s - loss: 0.0923 - iou_score:
0.8983 - f1-score: 0.9446
Epoch 21: val_loss did not improve from 0.17867
200/200 [==============================] - 113s 565ms/step - loss: 0.0923 -
iou_score: 0.8983 - f1-score: 0.9446 - val_loss: 0.1795 - val_iou_score: 0.8218
- val_f1-score: 0.8876 - lr: 1.0000e-05
Epoch 22/100
200/200 [==============================] - ETA: 0s - loss: 0.0866 - iou_score:
0.9036 - f1-score: 0.9477
Epoch 22: val_loss did not improve from 0.17867
200/200 [==============================] - 113s 566ms/step - loss: 0.0866 -
iou_score: 0.9036 - f1-score: 0.9477 - val_loss: 0.1794 - val_iou_score: 0.8222
- val_f1-score: 0.8879 - lr: 1.0000e-05
Epoch 23/100
200/200 [==============================] - ETA: 0s - loss: 0.0880 - iou_score:
0.9017 - f1-score: 0.9471
Epoch 23: val_loss did not improve from 0.17867
200/200 [==============================] - 113s 565ms/step - loss: 0.0880 -
iou_score: 0.9017 - f1-score: 0.9471 - val_loss: 0.1814 - val_iou_score: 0.8186
- val_f1-score: 0.8850 - lr: 1.0000e-05
Epoch 24/100
200/200 [==============================] - ETA: 0s - loss: 0.0849 - iou_score:
0.9055 - f1-score: 0.9494
Epoch 24: val_loss improved from 0.17867 to 0.17041, saving model to
model_weights/20221122-123004_segmentation_best_weights.hdf5
200/200 [==============================] - 115s 576ms/step - loss: 0.0849 -
iou_score: 0.9055 - f1-score: 0.9494 - val_loss: 0.1704 - val_iou_score: 0.8293
- val_f1-score: 0.8943 - lr: 1.0000e-05
```

```
Epoch 25/100
200/200 [==============================] - ETA: 0s - loss: 0.0843 - iou_score:
0.9067 - f1-score: 0.9499
Epoch 25: val_loss did not improve from 0.17041
200/200 [==============================] - 113s 566ms/step - loss: 0.0843 -
iou_score: 0.9067 - f1-score: 0.9499 - val_loss: 0.1765 - val_iou_score: 0.8231
- val_f1-score: 0.8889 - lr: 1.0000e-05
Epoch 26/100
200/200 [==============================] - ETA: 0s - loss: 0.0818 - iou_score:
0.9089 - f1-score: 0.9510
Epoch 26: val_loss did not improve from 0.17041
200/200 [==============================] - 113s 566ms/step - loss: 0.0818 -
iou_score: 0.9089 - f1-score: 0.9510 - val_loss: 0.1801 - val_iou_score: 0.8184
- val_f1-score: 0.8844 - lr: 1.0000e-05
Epoch 27/100
200/200 [==============================] - ETA: 0s - loss: 0.0807 - iou_score:
0.9095 - f1-score: 0.9515
Epoch 27: val_loss did not improve from 0.17041
200/200 [==============================] - 113s 566ms/step - loss: 0.0807 -
iou_score: 0.9095 - f1-score: 0.9515 - val_loss: 0.1752 - val_iou_score: 0.8263
- val_f1-score: 0.8914 - lr: 1.0000e-05
Epoch 28/100
200/200 [==============================] - ETA: 0s - loss: 0.0816 - iou_score:
0.9089 - f1-score: 0.9510
Epoch 28: val_loss did not improve from 0.17041
200/200 [==============================] - 113s 566ms/step - loss: 0.0816 -
iou_score: 0.9089 - f1-score: 0.9510 - val_loss: 0.1729 - val_iou_score: 0.8287
- val_f1-score: 0.8935 - lr: 1.0000e-05
Epoch 29/100
200/200 [==============================] - ETA: 0s - loss: 0.0812 - iou_score:
0.9078 - f1-score: 0.9504
Epoch 29: val_loss did not improve from 0.17041
200/200 [==============================] - 113s 565ms/step - loss: 0.0812 -
iou_score: 0.9078 - f1-score: 0.9504 - val_loss: 0.1772 - val_iou_score: 0.8233
- val_f1-score: 0.8888 - lr: 1.0000e-05
Epoch 30/100
200/200 [==============================] - ETA: 0s - loss: 0.0839 - iou_score:
0.9070 - f1-score: 0.9496
Epoch 30: val_loss did not improve from 0.17041
200/200 [==============================] - 113s 566ms/step - loss: 0.0839 -
iou_score: 0.9070 - f1-score: 0.9496 - val_loss: 0.1734 - val_iou_score: 0.8282
- val_f1-score: 0.8925 - lr: 1.0000e-05
Epoch 31/100
200/200 [==============================] - ETA: 0s - loss: 0.0780 - iou_score:
0.9131 - f1-score: 0.9534
Epoch 31: val_loss did not improve from 0.17041
200/200 [==============================] - 113s 566ms/step - loss: 0.0780 -
iou_score: 0.9131 - f1-score: 0.9534 - val_loss: 0.1751 - val_iou_score: 0.8260
```

```
- val_f1-score: 0.8907 - lr: 1.0000e-06
Epoch 32/100
200/200 [==============================] - ETA: 0s - loss: 0.0807 - iou_score:
0.9077 - f1-score: 0.9501
Epoch 32: val_loss did not improve from 0.17041
200/200 [==============================] - 114s 567ms/step - loss: 0.0807 -
iou_score: 0.9077 - f1-score: 0.9501 - val_loss: 0.1739 - val_iou_score: 0.8275
- val_f1-score: 0.8921 - lr: 1.0000e-06
Epoch 33/100
200/200 [==============================] - ETA: 0s - loss: 0.0818 - iou_score:
0.9087 - f1-score: 0.9508
Epoch 33: val_loss did not improve from 0.17041
200/200 [==============================] - 113s 566ms/step - loss: 0.0818 -
iou_score: 0.9087 - f1-score: 0.9508 - val_loss: 0.1740 - val_iou_score: 0.8269
- val_f1-score: 0.8918 - lr: 1.0000e-06
Epoch 34/100
200/200 [==============================] - ETA: 0s - loss: 0.0789 - iou_score:
0.9116 - f1-score: 0.9528
Epoch 34: val_loss did not improve from 0.17041
200/200 [==============================] - 113s 567ms/step - loss: 0.0789 -
iou_score: 0.9116 - f1-score: 0.9528 - val_loss: 0.1748 - val_iou_score: 0.8254
- val_f1-score: 0.8905 - lr: 1.0000e-06
Epoch 34: early stopping
```

[ ]:
```python
# 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)
```

```
20221122-123004
```

[ ]:
```
# 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")
```

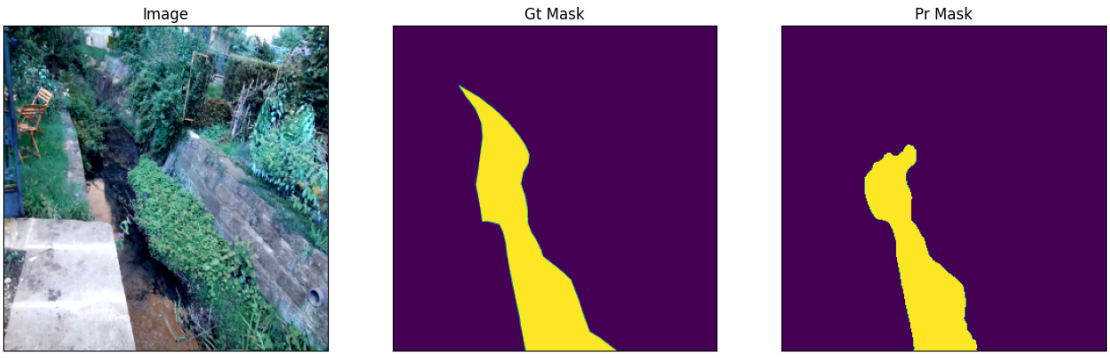
[ ]:
```
scores = model.evaluate(test_dataloader)

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

```
111/111 [==============================] - 7s 56ms/step - loss: -45.6569 -
iou_score: 0.8945 - f1-score: 0.8171
Loss: -45.657
mean iou_score: 0.8945
mean f1-score: 0.81714
```

5 Visualization of results on test dataset

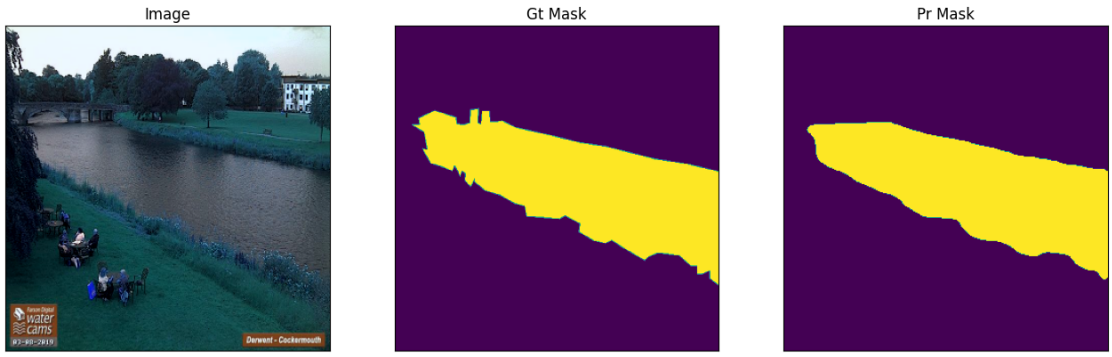
[ ]:
```
n = 5
ids = np.random.choice(np.arange(len(test_dataset)), size=n)

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 [==============================] - 7s 7s/step
```

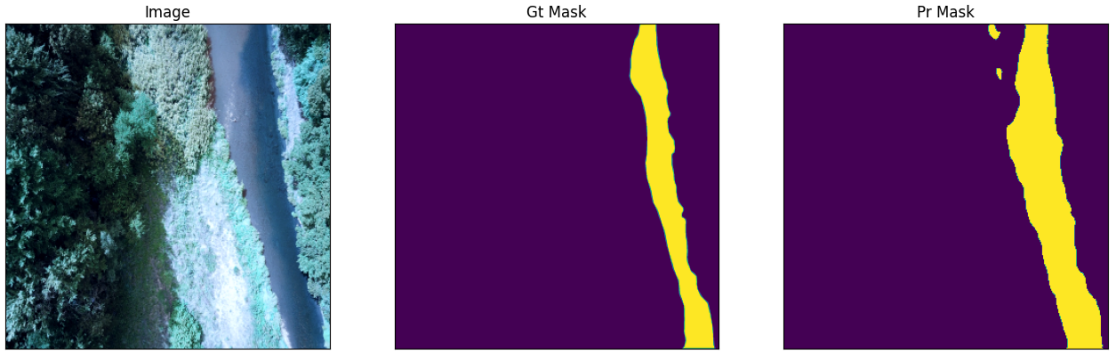



```
1/1 [==============================] - 0s 62ms/step
```

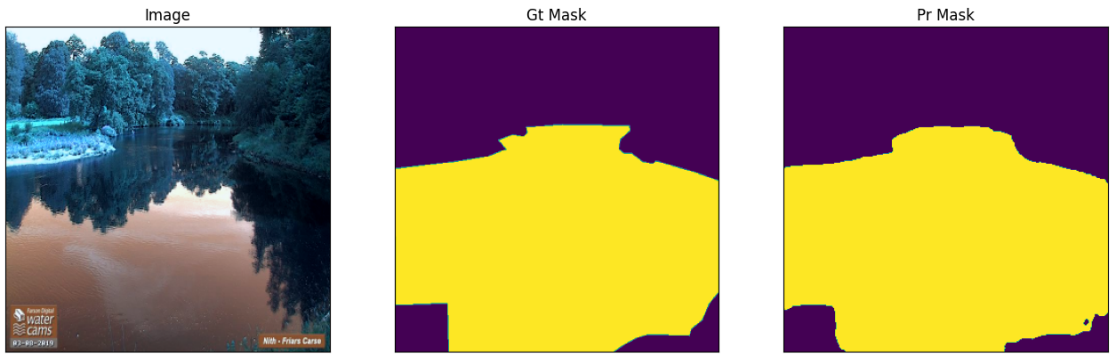

```
1/1 [==============================] - 0s 62ms/step
```




```
1/1 [==============================] - 0s 61ms/step
```




```
1/1 [==============================] - 0s 61ms/step
```

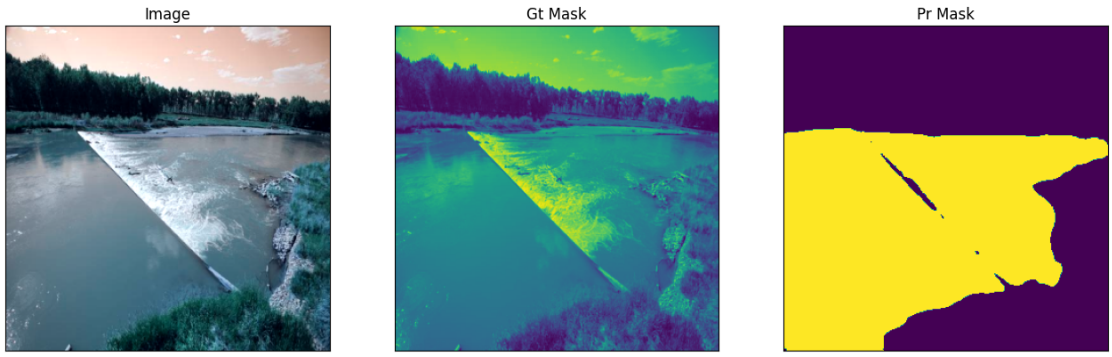


5.1 Visualize with test dataset of images for hydrology project

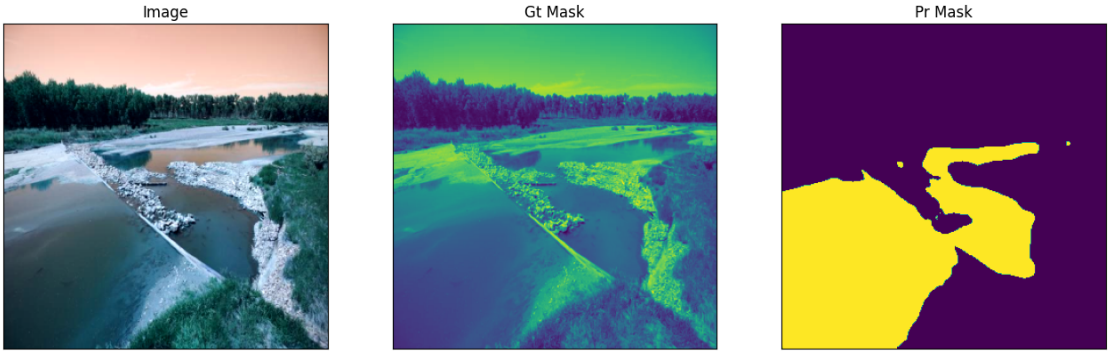
[ ]:
```
test_h_dataset = Dataset(
    #"river_segmented_hydrology/images",
    "dataset_testing",
    #"river_segmented_hydrology/masks",
    "dataset_testing",
    classes=CLASSES,
    #augmentation=get_image_fixing(),
    preprocessing=get_preprocessing(preprocess_input),
)

#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)

print(len(test_h_dataset))

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(),
    )
```

```
17
1/1 [==============================] - 0s 62ms/step
```

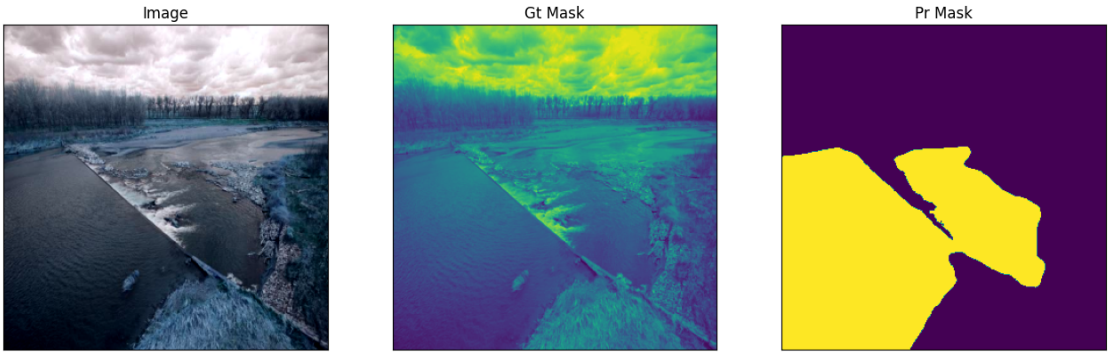



```
1/1 [==============================] - 0s 61ms/step
```

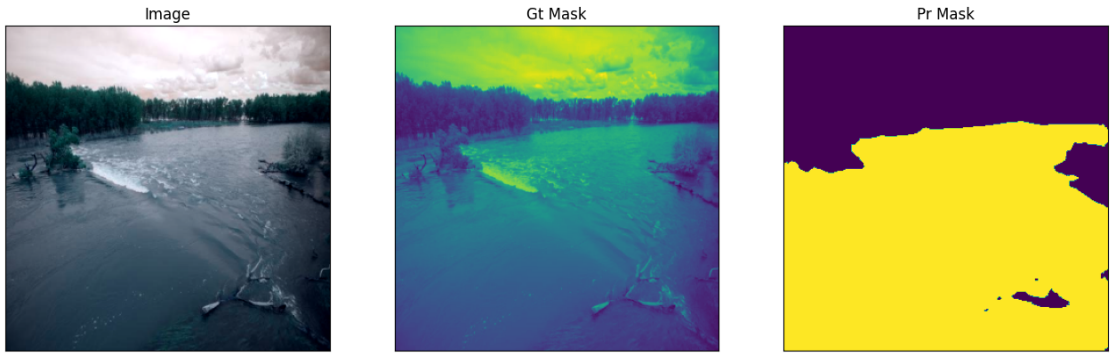



```
1/1 [==============================] - 0s 64ms/step
```

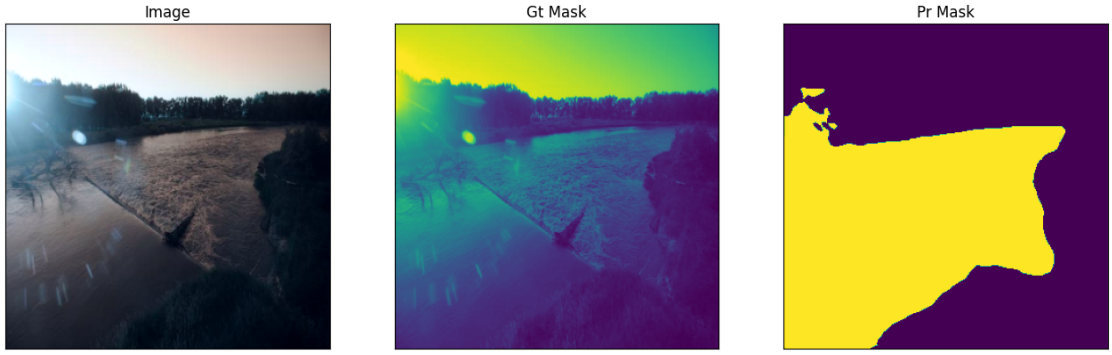

```
1/1 [==============================] - 0s 67ms/step
```



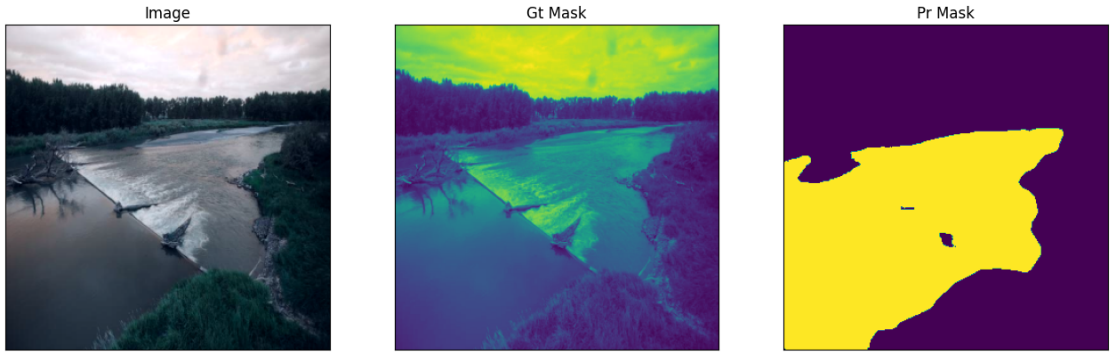

```
1/1 [==============================] - 0s 66ms/step
```



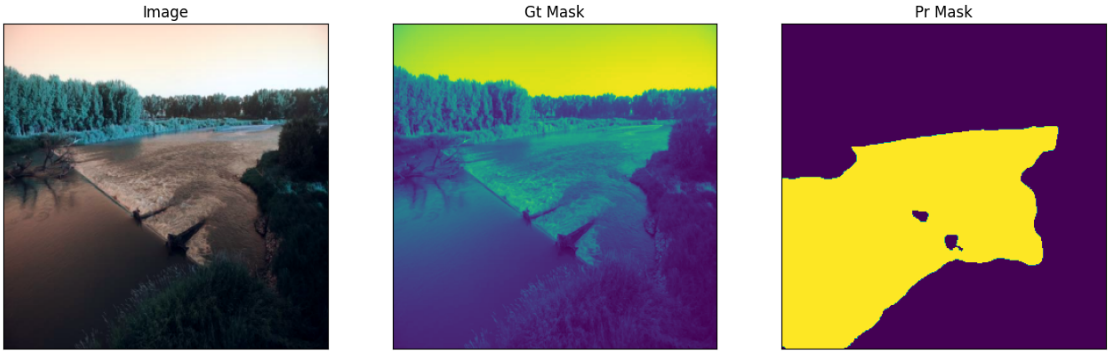

```
1/1 [==============================] - 0s 62ms/step
```



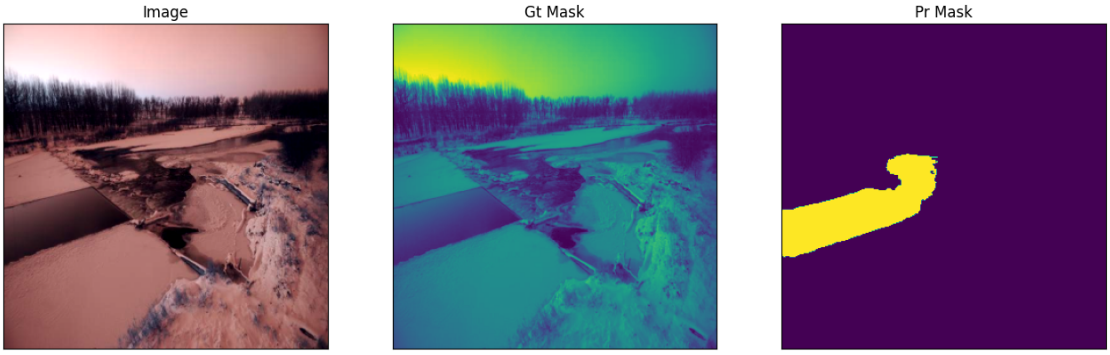

```
1/1 [==============================] - 0s 64ms/step
```



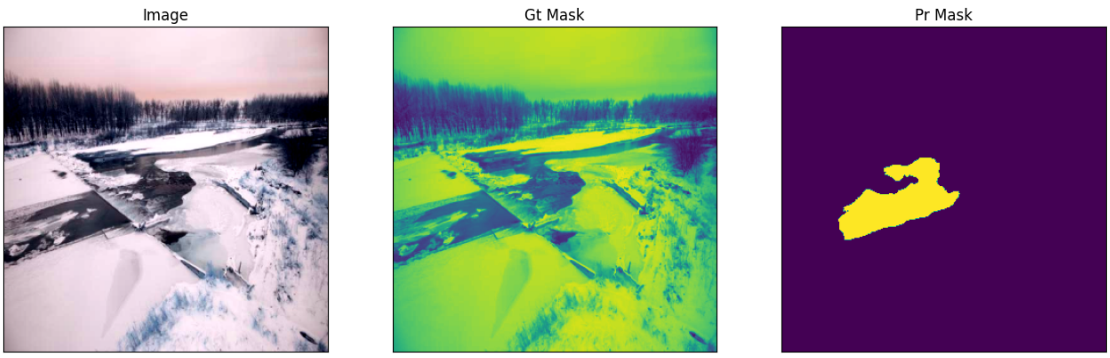

```
1/1 [==============================] - 0s 66ms/step
```



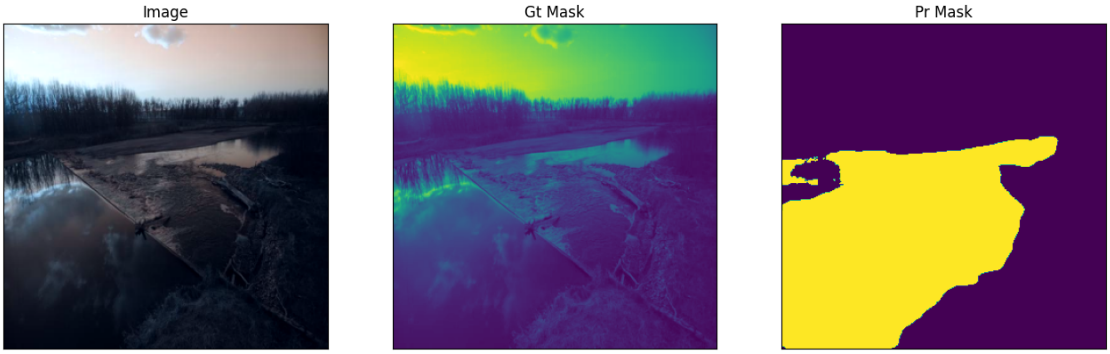

```
1/1 [==============================] - 0s 68ms/step
```



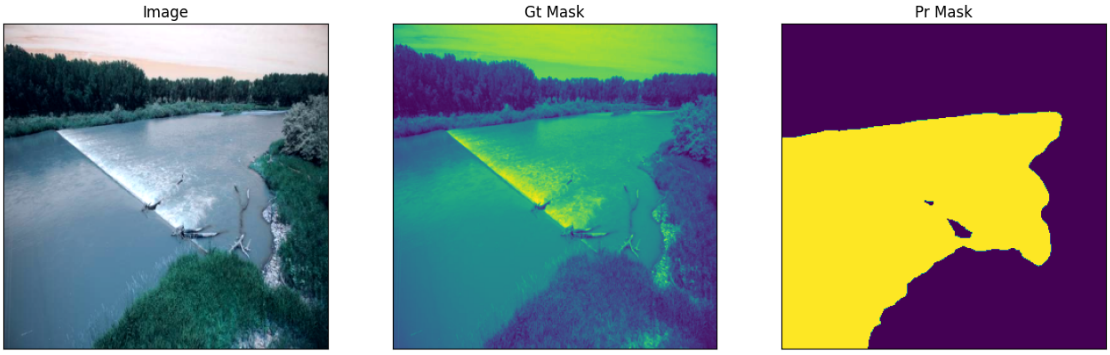

```
1/1 [==============================] - 0s 69ms/step
```



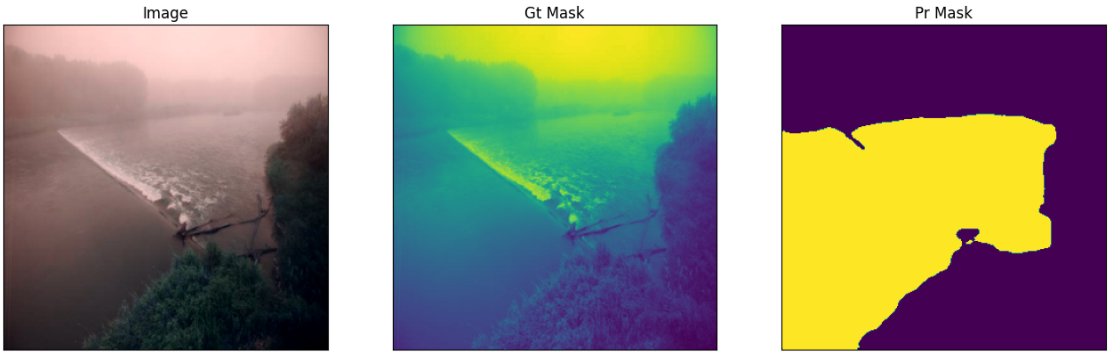

```
1/1 [==============================] - 0s 69ms/step
```



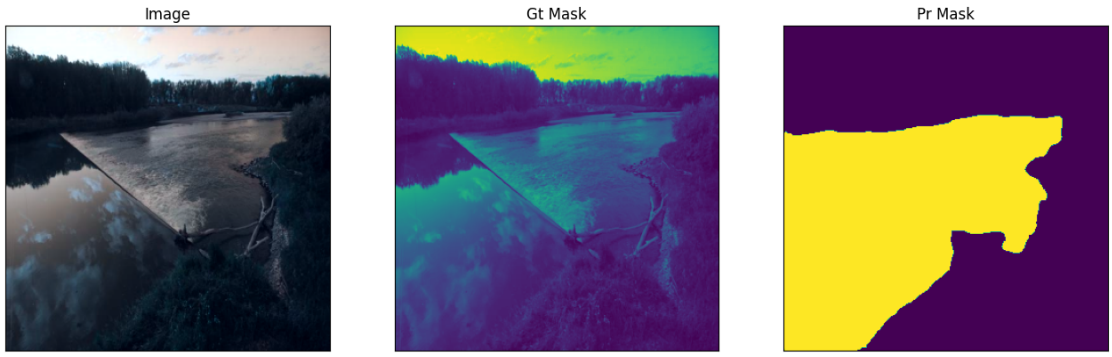

```
1/1 [==============================] - 0s 66ms/step
```



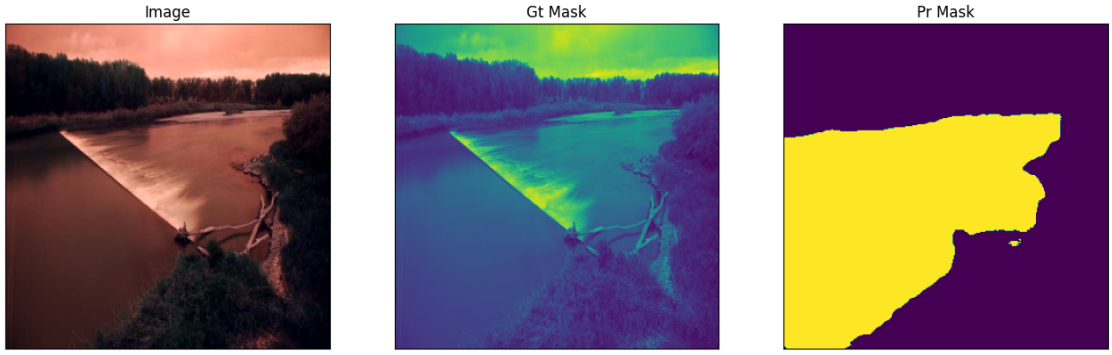

```
1/1 [==============================] - 0s 70ms/step
```



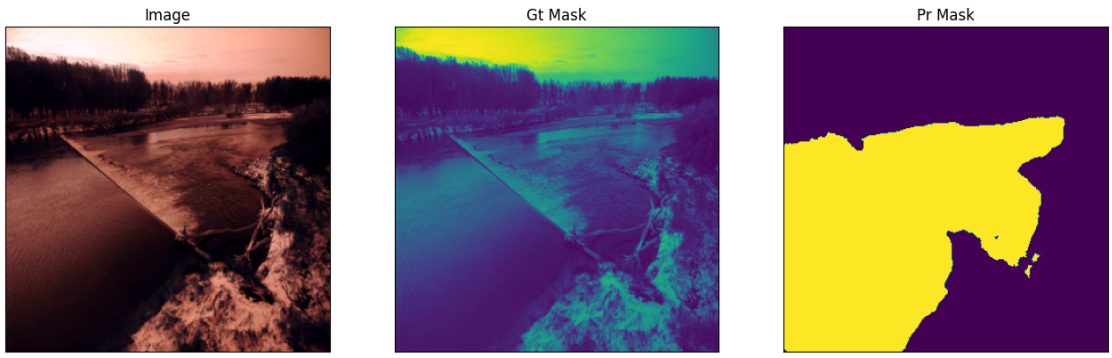

```
1/1 [==============================] - 0s 71ms/step
```




```
1/1 [==============================] - 0s 68ms/step
```




```
1/1 [==============================] - 0s 74ms/step
```




```
1/1 [==============================] - 0s 68ms/step
```

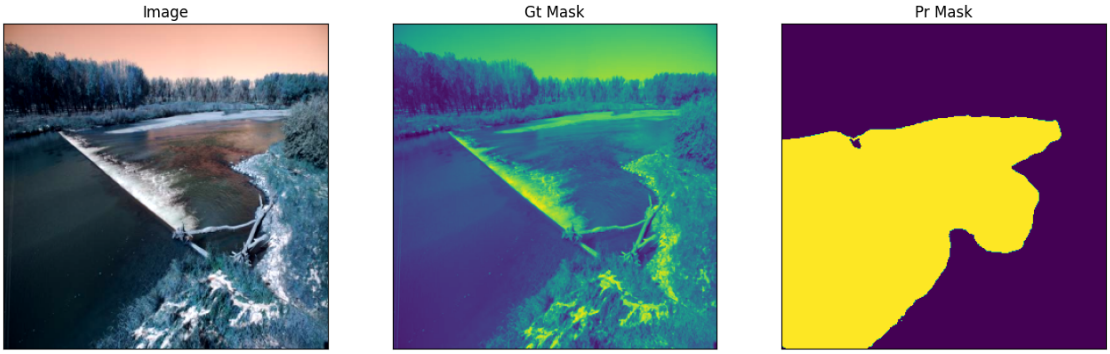


Image
Gt Mask
Pr Mask