

Подключение google disk для использования dataset

```
from google.colab import drive
drive.mount('/content/drive')
```

⇒ Drive already mounted at /content/drive; to attempt to forcibly remount, call

Установка библиотеки

```
!pip install petroscope
```

⇒ Requirement already satisfied: petroscope in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: pyyaml in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: numpy<2.0.0,>=1.16 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: pillow in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: matplotlib in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: scipy in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: loguru in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: prettytable in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: cycycler>=0.10 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: wcwidth in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages

Импорт необходимых библиотек

```

from pathlib import Path
from petroscope.segmentation.classes import ClassSet, LumenStoneClasses
from petroscope.segmentation.utils import load_image, load_mask
from petroscope.segmentation import GeoSegmModel
import numpy as np
from tqdm import tqdm
import tensorflow as tf
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Input, Conv2D, MaxPooling2D, UpSampling2D, cor
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.utils import to_categorical
from skimage.color import rgb2lab

```

Путь к dataset

```
ds_path = Path('/content/drive/MyDrive/PhotoSets/')
```

Инициализация набора классов для сегментации

```
classset = LumenStoneClasses.S1v1()
```

```

for cl in classset.classes:
    print(cl)

```

```

➡ [0, bg (background), color: #000000]
   [1, ccp/kub (chalcopyrite/cubanite), color: #ffa500]
   [2, gl (galena), color: #9acd32]
   [4, brt (bornite), color: #00bfff]
   [6, py/mrc (pyrite/marcasite), color: #2f4f4f]
   [8, sph (sphalerite), color: #ee82ee]
   [11, tnt/ttr (tenantite/tetrahedrite), color: #483d8b]

```

Формирование путей к изображениям и маскам

```

train_img_mask_p = [
    (img_p, ds_path / "masks" / "train" / f"{img_p.stem}.png")
    for img_p in sorted((ds_path / "imgs" / "train").iterdir())
]

test_img_mask_p = [
    (img_p, ds_path / "masks" / "test" / f"{img_p.stem}.png")
    for img_p in sorted((ds_path / "imgs" / "test").iterdir())
]

```

Загрузка и конвертация обучающих изображений в пространство LAB

```
for img_p, _ in train_img_mask_p:
    img = load_image(img_p, normalize=True)
    img_lab = rgb2lab(img)
    print(f"Image {img_p.name}: {img_lab.shape}, {img_lab.dtype}")
```

```
⇒ Image train_01.jpg: (2547, 3396, 3), float32
Image train_02.jpg: (2547, 3396, 3), float32
Image train_03.jpg: (2547, 3396, 3), float32
Image train_04.jpg: (2547, 3396, 3), float32
Image train_06.jpg: (2547, 3396, 3), float32
Image train_07.jpg: (2547, 3396, 3), float32
Image train_08.jpg: (2547, 3396, 3), float32
Image train_09.jpg: (2547, 3396, 3), float32
Image train_10.jpg: (2547, 3396, 3), float32
Image train_11.jpg: (2547, 3396, 3), float32
Image train_12.jpg: (2547, 3396, 3), float32
Image train_13.jpg: (2547, 3396, 3), float32
Image train_14.jpg: (2547, 3396, 3), float32
Image train_15.jpg: (2547, 3396, 3), float32
Image train_16.jpg: (2547, 3396, 3), float32
Image train_17.jpg: (2547, 3396, 3), float32
Image train_18.jpg: (2547, 3396, 3), float32
Image train_19.jpg: (2547, 3396, 3), float32
Image train_20.jpg: (2547, 3396, 3), float32
Image train_21.jpg: (2547, 3396, 3), float32
Image train_22.jpg: (2547, 3396, 3), float32
Image train_23.jpg: (2547, 3396, 3), float32
Image train_24.jpg: (2547, 3396, 3), float32
Image train_25.jpg: (2547, 3396, 3), float32
Image train_26.jpg: (2547, 3396, 3), float32
Image train_27.jpg: (2547, 3396, 3), float32
Image train_28.jpg: (2547, 3396, 3), float32
Image train_29.jpg: (2547, 3396, 3), float32
Image train_30.jpg: (2547, 3396, 3), float32
Image train_31.jpg: (2547, 3396, 3), float32
Image train_32.jpg: (2547, 3396, 3), float32
Image train_33.jpg: (2547, 3396, 3), float32
Image train_34.jpg: (2547, 3396, 3), float32
Image train_35.jpg: (2547, 3396, 3), float32
Image train_36.jpg: (2547, 3396, 3), float32
Image train_37.jpg: (2547, 3396, 3), float32
Image train_38.jpg: (2547, 3396, 3), float32
Image train_39.jpg: (2547, 3396, 3), float32
Image train_40.jpg: (2547, 3396, 3), float32
Image train_41.jpg: (2547, 3396, 3), float32
Image train_42.jpg: (2547, 3396, 3), float32
Image train_43.jpg: (2547, 3396, 3), float32
Image train_44.jpg: (2547, 3396, 3), float32
Image train_45.jpg: (2547, 3396, 3), float32
Image train_46.jpg: (2547, 3396, 3), float32
```

```
Image train_47.jpg: (2547, 3396, 3), float32
Image train_48.jpg: (2547, 3396, 3), float32
Image train_49.jpg: (2547, 3396, 3), float32
Image train_50.jpg: (2547, 3396, 3), float32
Image train_51.jpg: (2547, 3396, 3), float32
Image train_52.jpg: (2547, 3396, 3), float32
Image train_53.jpg: (2547, 3396, 3), float32
Image train_54.jpg: (2547, 3396, 3), float32
Image train_55.jpg: (2547, 3396, 3), float32
Image train_56.jpg: (2547, 3396, 3), float32
Image train_57.jpg: (2547, 3396, 3), float32
Image train_58.jpg: (2547, 3396, 3), float32
Image train_59.jpg: (2547, 3396, 3), float32
```

Загрузка масок без one-hot кодирования

```
for _, mask_p in train_img_mask_p:
    mask = load_mask(mask_p, classes=classset, one_hot=False)
    print(f"Mask {mask_p.name}: {mask.shape}, {mask.dtype}")
```

```
⇒ Mask train_01.png: (2547, 3396), uint8
Mask train_02.png: (2547, 3396), uint8
Mask train_03.png: (2547, 3396), uint8
Mask train_04.png: (2547, 3396), uint8
Mask train_06.png: (2547, 3396), uint8
Mask train_07.png: (2547, 3396), uint8
Mask train_08.png: (2547, 3396), uint8
Mask train_09.png: (2547, 3396), uint8
Mask train_10.png: (2547, 3396), uint8
Mask train_11.png: (2547, 3396), uint8
Mask train_12.png: (2547, 3396), uint8
Mask train_13.png: (2547, 3396), uint8
Mask train_14.png: (2547, 3396), uint8
Mask train_15.png: (2547, 3396), uint8
Mask train_16.png: (2547, 3396), uint8
Mask train_17.png: (2547, 3396), uint8
Mask train_18.png: (2547, 3396), uint8
Mask train_19.png: (2547, 3396), uint8
Mask train_20.png: (2547, 3396), uint8
Mask train_21.png: (2547, 3396), uint8
Mask train_22.png: (2547, 3396), uint8
Mask train_23.png: (2547, 3396), uint8
Mask train_24.png: (2547, 3396), uint8
Mask train_25.png: (2547, 3396), uint8
Mask train_26.png: (2547, 3396), uint8
Mask train_27.png: (2547, 3396), uint8
Mask train_28.png: (2547, 3396), uint8
Mask train_29.png: (2547, 3396), uint8
Mask train_30.png: (2547, 3396), uint8
Mask train_31.png: (2547, 3396), uint8
Mask train_32.png: (2547, 3396), uint8
Mask train_33.png: (2547, 3396), uint8
```

```
Mask train_34.png: (2547, 3396), uint8
Mask train_35.png: (2547, 3396), uint8
Mask train_36.png: (2547, 3396), uint8
Mask train_37.png: (2547, 3396), uint8
Mask train_38.png: (2547, 3396), uint8
Mask train_39.png: (2547, 3396), uint8
Mask train_40.png: (2547, 3396), uint8
Mask train_41.png: (2547, 3396), uint8
Mask train_42.png: (2547, 3396), uint8
Mask train_43.png: (2547, 3396), uint8
Mask train_44.png: (2547, 3396), uint8
Mask train_45.png: (2547, 3396), uint8
Mask train_46.png: (2547, 3396), uint8
Mask train_47.png: (2547, 3396), uint8
Mask train_48.png: (2547, 3396), uint8
Mask train_49.png: (2547, 3396), uint8
Mask train_50.png: (2547, 3396), uint8
Mask train_51.png: (2547, 3396), uint8
Mask train_52.png: (2547, 3396), uint8
Mask train_53.png: (2547, 3396), uint8
Mask train_54.png: (2547, 3396), uint8
Mask train_55.png: (2547, 3396), uint8
Mask train_56.png: (2547, 3396), uint8
Mask train_57.png: (2547, 3396), uint8
Mask train_58.png: (2547, 3396), uint8
Mask train_59.png: (2547, 3396), uint8
```

Загрузка масок с one-hot кодированием

```
for _, mask_p in train_img_mask_p:
    mask_one_hot = load_mask(mask_p, classes=classset, one_hot=True)
    print(f"Mask one-hot {mask_p.name}: {mask_one_hot.shape}, {mask_one_hot.dtype}")
```

```
⇒ Mask one-hot train_01.png: (2547, 3396, 7), float32
Mask one-hot train_02.png: (2547, 3396, 7), float32
Mask one-hot train_03.png: (2547, 3396, 7), float32
Mask one-hot train_04.png: (2547, 3396, 7), float32
Mask one-hot train_06.png: (2547, 3396, 7), float32
Mask one-hot train_07.png: (2547, 3396, 7), float32
Mask one-hot train_08.png: (2547, 3396, 7), float32
Mask one-hot train_09.png: (2547, 3396, 7), float32
Mask one-hot train_10.png: (2547, 3396, 7), float32
Mask one-hot train_11.png: (2547, 3396, 7), float32
Mask one-hot train_12.png: (2547, 3396, 7), float32
Mask one-hot train_13.png: (2547, 3396, 7), float32
Mask one-hot train_14.png: (2547, 3396, 7), float32
Mask one-hot train_15.png: (2547, 3396, 7), float32
Mask one-hot train_16.png: (2547, 3396, 7), float32
Mask one-hot train_17.png: (2547, 3396, 7), float32
Mask one-hot train_18.png: (2547, 3396, 7), float32
Mask one-hot train_19.png: (2547, 3396, 7), float32
Mask one-hot train_20.png: (2547, 3396, 7), float32
```

```
Mask one-hot train_21.png: (2547, 3396, 7), float32
Mask one-hot train_22.png: (2547, 3396, 7), float32
Mask one-hot train_23.png: (2547, 3396, 7), float32
Mask one-hot train_24.png: (2547, 3396, 7), float32
Mask one-hot train_25.png: (2547, 3396, 7), float32
Mask one-hot train_26.png: (2547, 3396, 7), float32
Mask one-hot train_27.png: (2547, 3396, 7), float32
Mask one-hot train_28.png: (2547, 3396, 7), float32
Mask one-hot train_29.png: (2547, 3396, 7), float32
Mask one-hot train_30.png: (2547, 3396, 7), float32
Mask one-hot train_31.png: (2547, 3396, 7), float32
Mask one-hot train_32.png: (2547, 3396, 7), float32
Mask one-hot train_33.png: (2547, 3396, 7), float32
Mask one-hot train_34.png: (2547, 3396, 7), float32
Mask one-hot train_35.png: (2547, 3396, 7), float32
Mask one-hot train_36.png: (2547, 3396, 7), float32
Mask one-hot train_37.png: (2547, 3396, 7), float32
Mask one-hot train_38.png: (2547, 3396, 7), float32
Mask one-hot train_39.png: (2547, 3396, 7), float32
Mask one-hot train_40.png: (2547, 3396, 7), float32
Mask one-hot train_41.png: (2547, 3396, 7), float32
Mask one-hot train_42.png: (2547, 3396, 7), float32
Mask one-hot train_43.png: (2547, 3396, 7), float32
Mask one-hot train_44.png: (2547, 3396, 7), float32
Mask one-hot train_45.png: (2547, 3396, 7), float32
Mask one-hot train_46.png: (2547, 3396, 7), float32
Mask one-hot train_47.png: (2547, 3396, 7), float32
Mask one-hot train_48.png: (2547, 3396, 7), float32
Mask one-hot train_49.png: (2547, 3396, 7), float32
Mask one-hot train_50.png: (2547, 3396, 7), float32
Mask one-hot train_51.png: (2547, 3396, 7), float32
Mask one-hot train_52.png: (2547, 3396, 7), float32
Mask one-hot train_53.png: (2547, 3396, 7), float32
Mask one-hot train_54.png: (2547, 3396, 7), float32
Mask one-hot train_55.png: (2547, 3396, 7), float32
Mask one-hot train_56.png: (2547, 3396, 7), float32
Mask one-hot train_57.png: (2547, 3396, 7), float32
Mask one-hot train_58.png: (2547, 3396, 7), float32
Mask one-hot train_59.png: (2547, 3396, 7), float32
```

Загрузка цветных масок

```
for img_p, _ in train_img_mask_p:
    mask_colored_path = ds_path / "masks_colored_png" / "train" / f"{img_p.stem}"
    mask_colored = load_image(mask_colored_path, normalize=False)
    print(f"Colored mask {mask_colored_path.name}: {mask_colored.shape}, {mask_c
```

```
→ Colored mask train_01.png: (2547, 3396, 3), uint8
   Colored mask train_02.png: (2547, 3396, 3), uint8
   Colored mask train_03.png: (2547, 3396, 3), uint8
   Colored mask train_04.png: (2547, 3396, 3), uint8
   Colored mask train_06.png: (2547, 3396, 3), uint8
```

```
Colored mask train_07.png: (2547, 3396, 3), uint8
Colored mask train_08.png: (2547, 3396, 3), uint8
Colored mask train_09.png: (2547, 3396, 3), uint8
Colored mask train_10.png: (2547, 3396, 3), uint8
Colored mask train_11.png: (2547, 3396, 3), uint8
Colored mask train_12.png: (2547, 3396, 3), uint8
Colored mask train_13.png: (2547, 3396, 3), uint8
Colored mask train_14.png: (2547, 3396, 3), uint8
Colored mask train_15.png: (2547, 3396, 3), uint8
Colored mask train_16.png: (2547, 3396, 3), uint8
Colored mask train_17.png: (2547, 3396, 3), uint8
Colored mask train_18.png: (2547, 3396, 3), uint8
Colored mask train_19.png: (2547, 3396, 3), uint8
Colored mask train_20.png: (2547, 3396, 3), uint8
Colored mask train_21.png: (2547, 3396, 3), uint8
Colored mask train_22.png: (2547, 3396, 3), uint8
Colored mask train_23.png: (2547, 3396, 3), uint8
Colored mask train_24.png: (2547, 3396, 3), uint8
Colored mask train_25.png: (2547, 3396, 3), uint8
Colored mask train_26.png: (2547, 3396, 3), uint8
Colored mask train_27.png: (2547, 3396, 3), uint8
Colored mask train_28.png: (2547, 3396, 3), uint8
Colored mask train_29.png: (2547, 3396, 3), uint8
Colored mask train_30.png: (2547, 3396, 3), uint8
Colored mask train_31.png: (2547, 3396, 3), uint8
Colored mask train_32.png: (2547, 3396, 3), uint8
Colored mask train_33.png: (2547, 3396, 3), uint8
Colored mask train_34.png: (2547, 3396, 3), uint8
Colored mask train_35.png: (2547, 3396, 3), uint8
Colored mask train_36.png: (2547, 3396, 3), uint8
Colored mask train_37.png: (2547, 3396, 3), uint8
Colored mask train_38.png: (2547, 3396, 3), uint8
Colored mask train_39.png: (2547, 3396, 3), uint8
Colored mask train_40.png: (2547, 3396, 3), uint8
Colored mask train_41.png: (2547, 3396, 3), uint8
Colored mask train_42.png: (2547, 3396, 3), uint8
Colored mask train_43.png: (2547, 3396, 3), uint8
Colored mask train_44.png: (2547, 3396, 3), uint8
Colored mask train_45.png: (2547, 3396, 3), uint8
Colored mask train_46.png: (2547, 3396, 3), uint8
Colored mask train_47.png: (2547, 3396, 3), uint8
Colored mask train_48.png: (2547, 3396, 3), uint8
Colored mask train_49.png: (2547, 3396, 3), uint8
Colored mask train_50.png: (2547, 3396, 3), uint8
Colored mask train_51.png: (2547, 3396, 3), uint8
Colored mask train_52.png: (2547, 3396, 3), uint8
Colored mask train_53.png: (2547, 3396, 3), uint8
Colored mask train_54.png: (2547, 3396, 3), uint8
Colored mask train_55.png: (2547, 3396, 3), uint8
Colored mask train_56.png: (2547, 3396, 3), uint8
Colored mask train_57.png: (2547, 3396, 3), uint8
Colored mask train_58.png: (2547, 3396, 3), uint8
Colored mask train_59.png: (2547, 3396, 3), uint8
```


Функция для загрузки и предобработки изображений и масок

```
def load_and_preprocess(img_path, mask_path, classes, img_size=(256, 256)):
    img = load_image(img_path, normalize=True)
    img = tf.image.resize(img, img_size)

    mask = load_mask(mask_path, classes=classes, one_hot=False)
    mask = tf.image.resize(mask[...], np.newaxis, img_size, method='nearest')
    mask = to_categorical(mask, num_classes=len(classes))

    return img, mask
```

Модель U-Net

```
def unet_model(input_size=(256, 256, 3), num_classes=7):
    inputs = Input(input_size)

    conv1 = Conv2D(64, 3, activation='relu', padding='same')(inputs)
    conv1 = Conv2D(64, 3, activation='relu', padding='same')(conv1)
    pool1 = MaxPooling2D(pool_size=(2, 2))(conv1)
    conv2 = Conv2D(128, 3, activation='relu', padding='same')(pool1)
    conv2 = Conv2D(128, 3, activation='relu', padding='same')(conv2)
    pool2 = MaxPooling2D(pool_size=(2, 2))(conv2)
    conv3 = Conv2D(256, 3, activation='relu', padding='same')(pool2)
    conv3 = Conv2D(256, 3, activation='relu', padding='same')(conv3)
    pool3 = MaxPooling2D(pool_size=(2, 2))(conv3)
    conv4 = Conv2D(512, 3, activation='relu', padding='same')(pool3)
    conv4 = Conv2D(512, 3, activation='relu', padding='same')(conv4)

    up5 = concatenate([UpSampling2D(size=(2, 2))(conv4), conv3], axis=-1)
    conv5 = Conv2D(256, 3, activation='relu', padding='same')(up5)
    conv5 = Conv2D(256, 3, activation='relu', padding='same')(conv5)
    up6 = concatenate([UpSampling2D(size=(2, 2))(conv5), conv2], axis=-1)
    conv6 = Conv2D(128, 3, activation='relu', padding='same')(up6)
    conv6 = Conv2D(128, 3, activation='relu', padding='same')(conv6)
    up7 = concatenate([UpSampling2D(size=(2, 2))(conv6), conv1], axis=-1)
    conv7 = Conv2D(64, 3, activation='relu', padding='same')(up7)
    conv7 = Conv2D(64, 3, activation='relu', padding='same')(conv7)

    outputs = Conv2D(num_classes, 1, activation='softmax')(conv7)

    model = Model(inputs=[inputs], outputs=[outputs])
    model.compile(optimizer=Adam(), loss='categorical_crossentropy', metrics=['a
    return model
```


Загрузка и предобработка обучающих данных

```
train_data = [load_and_preprocess(img_p, mask_p, classset) for img_p, mask_p in
train_images, train_masks = zip(*train_data)
train_images = np.array(train_images)
train_masks = np.array(train_masks)

print(f"Train images shape: {train_images.shape}")
print(f"Train masks shape: {train_masks.shape}")

model = unet_model(num_classes=len(classset.classes))

model.fit([train_images], [train_masks], batch_size=4, epochs=200, validation_sp
```

```
Epoch 20/200
13/13 ————— 3s 201ms/step - accuracy: 0.8073 - loss: 0.6289 -
Epoch 21/200
13/13 ————— 3s 203ms/step - accuracy: 0.7321 - loss: 0.8244 -
Epoch 22/200
13/13 ————— 3s 202ms/step - accuracy: 0.8266 - loss: 0.6130 -
Epoch 23/200
13/13 ————— 5s 203ms/step - accuracy: 0.8053 - loss: 0.6213 -
Epoch 24/200
13/13 ————— 3s 200ms/step - accuracy: 0.7949 - loss: 0.6361 -
Epoch 25/200
13/13 ————— 3s 200ms/step - accuracy: 0.8014 - loss: 0.6387 -
Epoch 26/200
13/13 ————— 5s 203ms/step - accuracy: 0.7743 - loss: 0.6786 -
Epoch 27/200
13/13 ————— 5s 198ms/step - accuracy: 0.8049 - loss: 0.6092 -
Epoch 28/200
13/13 ————— 5s 198ms/step - accuracy: 0.8270 - loss: 0.5513 -
Epoch 29/200
13/13 ————— 5s 199ms/step - accuracy: 0.7947 - loss: 0.6481 -
Epoch 30/200
13/13 ————— 3s 198ms/step - accuracy: 0.8052 - loss: 0.5963 -
Epoch 31/200
13/13 ————— 3s 200ms/step - accuracy: 0.7641 - loss: 0.7427 -
Epoch 32/200
13/13 ————— 3s 202ms/step - accuracy: 0.8194 - loss: 0.5694 -
Epoch 33/200
13/13 ————— 5s 201ms/step - accuracy: 0.8297 - loss: 0.5039 -
Epoch 34/200
13/13 ————— 5s 201ms/step - accuracy: 0.8036 - loss: 0.5538 -
Epoch 35/200
13/13 ————— 5s 208ms/step - accuracy: 0.8141 - loss: 0.5620 -
Epoch 36/200
13/13 ————— 5s 204ms/step - accuracy: 0.7864 - loss: 0.6319 -
Epoch 37/200
13/13 ————— 3s 202ms/step - accuracy: 0.8530 - loss: 0.4416 -
Epoch 38/200
13/13 ————— 5s 208ms/step - accuracy: 0.8140 - loss: 0.5360 -
Epoch 39/200
```

```

Epoch 39/200
13/13 _____ 3s 203ms/step - accuracy: 0.8019 - loss: 0.5998 -
Epoch 40/200
13/13 _____ 3s 202ms/step - accuracy: 0.7597 - loss: 0.6821 -
Epoch 41/200
13/13 _____ 3s 206ms/step - accuracy: 0.8315 - loss: 0.4722 -
Epoch 42/200
13/13 _____ 5s 208ms/step - accuracy: 0.8257 - loss: 0.5251 -
Epoch 43/200
13/13 _____ 5s 200ms/step - accuracy: 0.8369 - loss: 0.4990 -
Epoch 44/200
13/13 _____ 3s 204ms/step - accuracy: 0.8269 - loss: 0.5578 -
Epoch 45/200
13/13 _____ 5s 207ms/step - accuracy: 0.7640 - loss: 0.6551 -
Epoch 46/200
13/13 _____ 3s 199ms/step - accuracy: 0.8481 - loss: 0.5259 -
Epoch 47/200
13/13 _____ 5s 198ms/step - accuracy: 0.8200 - loss: 0.5771 -
Epoch 48/200
13/13 _____ 5s 204ms/step - accuracy: 0.8232 - loss: 0.5456 -
Epoch 49/200
13/13 _____ 3s 204ms/step - accuracy: 0.8410 - loss: 0.4780 -
Epoch 50/200

```

Тестирование

```

from petroscope.segmentation.eval import SegmDetailedTester
from tensorflow.keras.utils import to_categorical

tester = SegmDetailedTester(
    Path("output"),
    classes=classset,
    void_pad=0,
    void_border_width=4,
    vis_plots=False,
    vis_segmentation=True,
)

for img_p, mask_p in test_img_mask_p:
    img = load_image(img_p, normalize=True)
    img_resized = tf.image.resize(img, (256, 256))

    pred = model.predict(np.expand_dims(img_resized, axis=0))
    pred = np.argmax(pred[0], axis=-1)
    pred = pred.astype(np.uint8)
    print(f"Pred shape: {pred.shape}")

    mask = load_mask(mask_p, classes=classset, one_hot=False)
    mask_resized = tf.image.resize(mask[...], np.newaxis, (256, 256), method='nearest')
    mask_resized = mask_resized[...].numpy().astype(np.uint8)
    print(f"Mask resized shape: {mask_resized.shape}")

```

```

pred_one_hot = to_categorical(pred, num_classes=len(classset.classes))
mask_one_hot = to_categorical(mask_resized, num_classes=len(classset.classes))
print(f"Pred one-hot shape: {pred_one_hot.shape}")
print(f"Mask one-hot shape: {mask_one_hot.shape}")

metrics = tester.eval.evaluate(pred_one_hot, gt=mask_one_hot)
metrics_void = tester.eval_void.evaluate(pred_one_hot, gt=mask_one_hot)

print(f"Metrics for {img_p.name}:\n{metrics}")
print(f"Metrics with void borders for {img_p.name}:\n{metrics_void}")
print("-" * 50)

```

```

➔ /usr/local/lib/python3.11/dist-packages/keras/src/models/functional.py:237: l
Expected: ['keras_tensor']
Received: inputs=Tensor(shape=(1, 256, 256, 3))
  warnings.warn(msg)
1/1 ————— 4s 4s/step
Pred shape: (256, 256)
Mask resized shape: (256, 256)
Pred one-hot shape: (256, 256, 7)
Mask one-hot shape: (256, 256, 7)
Metrics for test_01.jpg:
  iou [soft]:
    bg: 0.8588 [0.8588]
    brt: 0.0000 [0.0000]
    ccp/kub: 0.0692 [0.0692]
    gl: 0.0408 [0.0408]
    py/mrc: 0.9151 [0.9151]
    sph: 0.7263 [0.7263]
    tnt/ttr: 0.0000 [0.0000]
  mean iou [soft]: 0.3729 [0.3729]
  acc: 0.8969

Metrics with void borders for test_01.jpg:
  iou [soft]:
    bg: 0.8588 [0.8588]
    brt: 0.0000 [0.0000]
    ccp/kub: 0.0692 [0.0692]
    gl: 0.0408 [0.0408]
    py/mrc: 0.9151 [0.9151]
    sph: 0.7263 [0.7263]
    tnt/ttr: 0.0000 [0.0000]
  mean iou [soft]: 0.3729 [0.3729]
  acc: 0.8969

```

```

-----
1/1 ————— 0s 59ms/step
Pred shape: (256, 256)
Mask resized shape: (256, 256)
Pred one-hot shape: (256, 256, 7)
Mask one-hot shape: (256, 256, 7)
Metrics for test_02.jpg:
  iou [soft]:

```

```
bg: 0.6540 [0.6540]
brt: 1.0000 [1.0000]
ccp/kub: 0.0000 [0.0000]
gl: 0.0069 [0.0069]
py/mrc: 0.0014 [0.0014]
sph: 0.0461 [0.0461]
tnt/ttr: 0.0000 [0.0000]
mean iou [soft]: 0.2441 [0.2441]
acc: 0.3457
```

Metrics with void borders for test_02.jpg:

iou [soft]:

```
bg: 0.6540 [0.6540]
brt: 1.0000 [1.0000]
ccp/kub: 0.0000 [0.0000]
gl: 0.0069 [0.0069]
py/mrc: 0.0014 [0.0014]
snh: 0.0461 [0.0461]
```

